



Escola Politècnica Superior
d'Enginyeria de Vilanova i la Geltrú

UNIVERSITAT POLITÈCNICA DE CATALUNYA

TREBALL FINAL DE GRAU **ACONDICIONAMENT D'UNA COLLITADORA MITJANÇANT SISTEMA ARDUINO**

TÍTOL: Acondicionament d'una collitadora mitjançant sistema Arduino

AUTORS: Casalí Sintes, Margarita
Guasch Cano, Ferran

TUTOR: Miret Tomás, Jaume

DATA: Juliol, 2018

COGNOMS: CASALÍ SINTES

NOM: MARGARITA

TITULACIÓ: GRAU EN ENGINYERIA MECÀNICA

PLA:

DIRECTOR: Miret Tomás, Jaume

DEPARTAMENT: Enginyeria Electrònica

COGNOMS: GUASCH CANO

NOM: FERRAN

TITULACIÓ: GRAU EN ENGINYERIA ELECTRONICA I AUROMÀTICA INDUSTRIAL

PLA:

DIRECTOR: Miret Tomás, Jaume

DEPARTAMENT: Enginyeria Electrònica

QUALIFICACIÓ DEL TFG

TRIBUNAL

PRESIDENT
Castilla Fernandez, Miguel

SECRETARI
Gaya Suñer, Pedro Francisco

VOCAL
Segalas Coral, Jordi

DATA DE LECTURA: 12 de Juliol del 2018

Aquest Projecte té en compte aspectes mediambientals: ☐ Sí ☐ No

RESUM

Mitjançant un microcontrolador molt comú en el mercat i d'un preu molt econòmic es realitzarà un sistema de sensorat per una collitadora. Aquest sistema conta amb un sensor de pressió, un de temperatura i 8 d'inductius. Per una banda, els dos primers ens serviran per parametritzar, com el seu nom indica, la pressió i temperatura del motor dièsel de la màquina. Per altra banda, els 8 sensors inductius tindran la funció de controlar les revolucions de gir dels 8 motors que componen la collitadora.

Per tal que l'usuari pugui assabentar-se quan un paràmetre dels controlats no és adequat, i per tant la màquina no funciona com s'espera, saltarà una alarma. Aquesta alarma serà visual, ja que apareixerà per pantalla, i també sonora.

El sistema, a més, contarà amb un mode configuració. Quan l'usuari del sistema vulgui canviar algun paràmetre dels que controla, podrà fer-ho mitjançant la connexió amb un PC, amb un progràma anomenat Putty, i per tant podrà canviar quan tenen que saltar les alarmes.

Paraules clau (màxim 10):

Microcontrolador	Sistema de Sensorat	Arduino		
Putty	Alarma	VGA		

ABSTRACT

Using one of the most common and cheaper microcontroller, we develop a sensor system for a harvester. This system has an eight inductive sensors, one temperature sensor and one pressure sensor. The last two are connect in the diesel motor of the harvester.

By the way, the other 8 inductive sensors have the function of control the rotation revolutions of the eight motors that compose the harvester.

So that the user can find out when a controlled parameter is not correct and therefore the machine does not work as good as expected, an alarm will appear. This alarm is visual, a message will appear on the screen explaining the alarm, and have sound to call the attention of the user.

The system will have with a configuration mode. When the user wants to change any parameter, he can do it, connecting the system to the PC and opening a program called PUTTY, with that he can change any parameter and then continues working with the harvester.

Keywords (10 maximum):

Microcontroller	Sensor System	Arduino		
Putty	Alarm	VGA		

APORTACIÓ INDIVIDUAL AL GRUP

El projecte, com es pot veure en aquesta memòria, es divideix en dos grans parts, Software i Hardware. Cada membre del grup s'ha fet responsable d'una d'aquestes parts, Margarita Casali de de la part Hardware i Ferran Guasch de la part de Software.

De totes maneres, els dos estudiants hem realitzat feina conjunta, per tal de poder resoldre els problemes que han sorgint a cada una de les parts a mesura que es desenvolupava el treball.

SUMARI

1.INTRODUCCIÓ	10
2.OBJECTIUS.....	11
3. DESCRIPCIÓ GENERAL DEL SISTEMA	12
4. HARDWARE	18
4.1 ARDUINO.....	18
4.2 SENSORS I ADAPTACIÓ DELS CIRUCITS	19
4.2.1 SENSOR DE TEMPERATURA	19
4.2.1.1 SENSOR ELEGIT : NI200	20
4.2.1.2 ADAPTACIÓ DEL SENSOR TEMPERATURA	21
4.2.2 SENSOR DE PRESSIÓ	27
4.2.2.1 SENSOR ELEGIT: WIKA-10A	28
4.2.2.1.2 ADAPTACIÓ DEL SENSOR DE PRESSIÓ	29
4.2.3 SENSORS INDUCITUS	34
4.2.3.1 SENSOR INDUCTIU ELEGIT: IME12-04BPSZCOS ^[17]	35
4.2.3.2 ADAPTACIÓ DEL SENSOR	36
4.3 SISTEMA DE ALARMA.....	38
4.3.1 ALARMA VISUAL	38
4.3.1.1 VGA.....	39
4.3.2 ALARMA SONORA.....	40
4.4 PLACA.....	43
5. SOFTWARE	46
5.1 PRINCIPAL FUNCIO DEL PROGRAMA	49
5.2 LLIBRERIES.....	53
5.2.1 ARDUINO TVOout	53
5.2.2 VGA DUE.....	53
5.2.3 DueTimer	55
5.3 FUNCIONS	56

5.3.1 FUNCIO DE CONFIGURACIO.....	56
5.3.2 FUNCIO RUN	58
5.3.3 FUNCIONS PER GUARDAR VARIABLES NO VOLATILS A LA MEMORIA FLASH	62
5.3.4 FUNCIONS DE LES ALARMES.....	62
5.4 CONNEXIO DEL TERMINAL	66
5.4.1 CONFIGURACIO PUTTY	66
6. MANUAL DE FUNCIONAMENT DEL SISTEMA PER L'USUARI.....	69
7. TEST DEL SISTEMA	84
7.1. PROGRAMA.....	84
7.1.1. MODE CONFIGURACIO	84
7.1.2. MODE RUN	86
7.2. CONEXIONS I SENSORS	87
8. PRESSUPOST ECÒNOMIC.....	89
9. CONCLUSIONS	90
10. AGRAÏMENTS.....	91
11. BIBLIOGRAFIA	92
11.1 ENTRADES DE FÒRUM:	92
11.2 ENTRADES DE BLOG:	92
11.3 E-BOOK:	93
11.4 VÍDEOS:	93
11.5 ARXIVS DESCARREGABLES:	93
11.6 WEBS.....	96
12. FOTOTECA	97
12.1 FIGURES	97
12.2 GRÀFIQUES	98
12. ANNEXES	99

SUMARI DE FIGURES

FIGURA 1: COLLITADORA MODEL LAVERDA M84.	12
FIGURA 2: ESQUEMA DE LA COLLITADORA, AMB TOTS ELS ELEMENTS QUE LA CONFORMEN.	14
FIGURA 3: COL·LOCACIÓ DELS DIFERENTS SENSORS A LA COLLITADORA.	15
FIGURA 4: DIAGRAMA DE BLOCS DEL SISTEMA	17
FIGURA 5: ARDUINO DUE	18
FIGURA 6: SENSOR DE TEMPERATURA, PT100.....	19
FIGURA 7: SENSOR DE TEMPERATURA, NI200	19
FIGURA 8. SENSOR DE TEMPERATURA, TERMOPAR DE TIPUS K I MODEL 363-0294.....	19
FIGURA 9: CIRCUIT D'ADAPTACIÓ DEL SENSOR DE TEMPERATURA NI200	26
FIGURA 10. SENSOR DE PRESSIÓ, MODEL DIN43650C, DE LA CASA GEMSSENSOR.....	28
FIGURA 11: SENSOR DE PRESSIÓ WIKA-A10	28
FIGURA 12: SENSOR DE PRESSIÓ DE LA CASA IFM, MODEL PT5415.	28
FIGURA 13: CIRCUIT D'ADAPTACIÓ DEL SENSOR DE PRESSIÓ.....	33
FIGURA 14: SENSOR INDUCTIU, MODEL IME12-04BPSZCOS	35
FIGURA 15: SENSOR INDUCTIU DE LA CASA PEPPERL+FUCHS, I MODEL NBB5-18GM5-E2-V1.....	35
FIGURA 16: SENSOR INDUCTIU DE LA CASA MODEL BES M08EC-PSC20B-S49G, DE LA CASA BALLUFF	35
FIGURA 17: CIRCUIT D'ADAPTACIÓ DELS SENSORS INDUCTIUS.....	36
FIGURA 18: CONNEXIÓ DEL DB-15 AMB EL DB-9	39
FIGURA 19: CONNEXIÓ DEL DB-9 AMB L'ARDUINO DUE	40
FIGURA 20: CIRCUIT D'ADAPTACIÓ DE L'ALARMA SONORA, AMB EL BUZZER.....	41
FIGURA 21: CIRCUIT GENERAL DEL SISTEMA	44
FIGURA 22: UBICACIÓ DELS ELEMENTS A LA PLACA PCB	45
FIGURA 23: LAYOUT PLACA CARA SUPERIOR	46
FIGURA 24: LAYOUT PLACA CARA INFERIOR	47
FIGURA 25: PLACA PCB.....	48
FIGURA 26: DIAGRAMA DE FLUX DEL PROGRAMA PRINCIPAL	52
FIGURA 27: DIAGRAMA DE FLUX DE LA FUNCIÓ LECTURA.	56
FIGURA 28: IMPLEMENTACIÓ DE LA FUNCIÓ "LECTURA".	57
FIGURA 29: IMPLEMENTACIÓ DE LA FUNCIÓ "RESETVALOR"	57
FIGURA 30: IMPLEMENTACIÓ DE LA FUNCIÓ INTMOTOR X ()	58
FIGURA 31: IMPLEMENTACIÓ DE LA FUNCIÓ "TIMERINT".....	59
FIGURA 32: IMPLEMENTACIÓ DE LA FUNCIÓ "REVOLUCIONS"	60

FIGURA 33: IMPLEMENTACIÓ DE LA FUNCIO "TEMPERATURA"	61
FIGURA 34:IMPLEMENTACIÓ DE LA FUNCIO "PRESSIÓ"	61
FIGURA 35: IMPLEMENTACIÓ DE LA FUNCIO PER GUARDAR DADES A LA MEMÒRIA FLASH.....	62
FIGURA 36: IMPLEMENTACIÓ DE LA FUNCIO PER MOSTRAR ELS VALORS PER PANTALLA.....	63
FIGURA 37: IMPLEMENTACIÓ DE LA FUNCIO DE L'ALARMA SONORA.....	64
FIGURA 38: IMPLEMENTACIÓ DE LA FUNCIO DE LECTURA DE LA MEMÒRIA FLASH.	65
FIGURA 39: CONNEXIÓ DEL PORT COM DE L'ARDUINO.....	67
FIGURA 40: GUARDAR SA CONFIGURACIÓ DE L'ARDUINO.	68
FIGURA 41: CONNEXIÓ DELS SENSORS A LA PLACA PCB.....	72
FIGURA 42: PLACA PCB AMB EL MICROCONTROLADOR ARDUINO	73
FIGURA 43: LLOC DE DESCAREGA DEL PUTTY.	74
FIGURA 44: ARXIU DESCAREGAT DEL PUTTY.....	75
FIGURA 45: PAS 1 DE LA CONFIGURACIÓ DEL PUTTY.....	75
FIGURA 46: PAS 2 DE LA CONFIGURACIÓ DEL PUTTY.....	76
FIGURA 47: GUARDAT DEL PERFIL DE L'ARDUINO	77
FIGURA 48: MODE DE CONFIGURACIÓ	78
FIGURA 49: IMATGE DEL MONITOR, AMB EL MISSATGE QUE VISUALITZARÀ L'USUARI	80
FIGURA 50: PÀGINA WEB PER DESCARGAR L'ARDUINO	81
FIGURA 51: CONFIGURACIÓ DE LA PLACA AL PROGRAMA	82
FIGURA 52: ICONOS DE COMPILAR I CARGAR EL PROGRAMA.....	83
FIGURA 53: PROGRAMA DE PROVA DE LECTURA.	84
FIGURA 54: PROGRAMA PROVA DE CONNEXIÓ AMB EL CLIENT PUTTY.	85
FIGURA 55: PROGRAMA PROVA DELS SENSORS DE TEMPERATURA I PRESSIÓ.	86
FIGURA 56: CONNEXIONS DEL PROGRAMA PROVA DEL COMPTA REVOLUCIONS.	87
FIGURA 57: MONTATGE DEL MONITOR.....	88
FIGURA 58: IMATGE DELS MISSATGES QUE SORTIRAN PEL MONITOR.....	88

SUMARI DE GRÀFICS

GRÀFIC 1: RELACIÓ DE TEMPERATURA-RESISTÈNCIA DE LA NI200.....	20
GRÀFIC 2: RELACIÓ DE TEMPERATURA-VOLTATGE ENTRADA AMPLIFIADOR.....	22
GRÀFIC 3: RELACIÓ TEMPERATURA- VOLTATGE QUE ARRIBA AL PIN DE L'ARDUINO.	25
GRÀFIC 4: RELACIÓ DE PRESSIÓ I CORRENT DE SORTIDA DEL SENSOR WIKA-A10.	29
GRÀFIC 5: RELACIÓ PRESSIÓ- TENSIO EN R1.	30
GRÀFIC 6: RELACIÓ PRESSIÓ- VOLTATGE QUE ARRIBA AL PIN A3 DE L'ARDUINO.	32

1.INTRODUCCIÓ

En el món de l'agricultura un dels moments més importants de l'any és el temps de recollida. Els pagesos i la gent que es dedica a l'agricultura saben bé que durant aquells dies és important el treball que realitzen, ja que si no es pot fer mal bé la feina de mesos.

Amb els avanços tecnològics, han arribat moltes millores que permeten fer la collita amb més eficiència i seguretat. Una d'aquestes millores són els sistemes de sensorat i monitorització de dades a temps real de diferents paràmetres de la maquinària, fent així que aquesta no pateixi problemes tècnics greus que puguin causar una aturada de la màquina i per tant que no es pugui dur a terme la collita.

Actualment, les collitadores que hi ha al mercat ja porten incorporat aquest tipus de sistema, igual que poden portar GPS, o altres complements. Però pagesos que realitzen aquesta feina com a segona ocupació, sovint, tenen maquinària més antiga, la qual no porta aquestes millores incorporades. Així doncs, aquest tipus d'usuaris han de confiar amb la seva experiència per saber quin problema té la màquina, o amb el soroll que aquesta fa saber si s'ha espatllat algun element mòbil. Aquest fet provoca poca precisió i que sigui molt més fàcil que una incidència petita no es detecti a temps i acabi sent una avaria més gran.

Una altra alternativa és incloure a la maquinària antiga un sistema de sensor i monitoratge de dades modern, que també té un preu elevat.

Per aquests motius, es pretén realitzar un sistema de sensors que pugui ser col·locat a una collitadora antiga, però a un preu molt més assequible que els que avui dia podem trobar al mercat.

2.OBJECTIUS

El propòsit d'aquest projecte és la implementació d'un sistema de sensors a una collitadora del model Laverda M84, mitjançant un sistema Arduino, que ens permeti detectar quan la màquina falla.

Aquest sistema disposa de 8 sensors inductius que aniran col·locats a llocs estratègics de la màquina, ens permetran saber a quina velocitat giren les diferents parts mòbils. Per altra banda, també és monitoritzaran els paràmetres de pressió i temperatura del motor dièsel de la mateixa màquina.

Per tal que l'operari de la collitadora s'assabenti d'un possible problema de la màquina o que una de les variables que controlen els diferents sensors no és adequat, apareixerà per pantalla un text d'avís, amb quin paràmetre falla i per què. D'igual manera, per tal de fer-ho més visible, també disposarà d'una alarma sonora, que alertarà del no correcte funcionament de la màquina.

Com ja hem mencionat, tot el tractament de dades que ens proporcionen els diferents sensors es realitzarà mitjançant un microcontrolador Arduino, ja que és una de les maneres més econòmiques de fer-ho.

A més per tal que els paràmetres que el nostre sistema controlarà puguin ser modificats segons el que l'usuari desitgi, també tindrà un mode de configuració. És a dir, els paràmetres del sistema podran ser modificats per ajustar en quin punt es vol que les alarmes s'activin, o quina velocitat és l'adequada pels motors.

Els objectius d'aquest projecte són varis. Per una banda, el disseny de circuits pel correcte funcionament dels sensors i l'adaptació d'aquests per l'Arduino, per tal d'obtenir un bon tractament de les dades. Un cop realitzat aquest treball, el disseny de la placa impresa (PCB). Per altra banda, la realització del codi Arduino.

3. DESCRIPCIÓ GENERAL DEL SISTEMA

La collitadora per la qual realitzem el sistema és un model Laverda M84, com la que és mostra a la figura 1. Aquesta collitadora, al ser antiga no porta cap tipus de sistema que permet la detecció d'avaries.



Figura 1: Collitadora model Laverda M84.

A dia d'avui, al mercat no hi ha sistemes de sensors com el que volem realitzar per adaptar una collitadora tan antiga com la que nosaltres disposem. Però, seguint els preus de mercat, si aquesta existís, rondaria un preu d'uns 4 o 5 mil €. El que si podem trobar avui dia, són sistemes de detecció de pèrdua de gra, els quals poden estar sobre els 2.500 €. Aquest tipus de sistemes són molt sofisticats i complets.

Per altra banda, com ja hem mencionat, la collitadora, Laverda M84, és una collitadora petita. Avui dia, una collitadora una mica més gran, amb un tall de 3, 6, o 4.2m, té un cost d'uns 120-200 mil € depenent de la qualitat d'aquesta, encara que podem trobar-ne de fins a 350 mil €.

Així doncs, com ja hem explicat, el que es pretén és realitzar un sistema de sensors que ens permeti analitzar tots els paràmetres que són importants pel seu correcte funcionament, sense arribar als preus de mercat.

Per entendre millor la funció que realitzarà aquest sistema, primerament haurem de realitzar una descripció general i un estudi del funcionament d'una collitadora de recollida de cereal, com és el nostre cas.

Aquest tipus de maquinària destaca sobretot per la recollida tant de gra com de cereal. Per poder recollir aquest tipus d'aliment, si ens guiem amb la figura 2, primerament el molinet (1), empeny la tija del cereal cap a la barra de sega (2), la qual serà l'encarregada de tallar-la. Un cop tallades, són dipositades sobre la plataforma del conductor transversal o alimentador (4), el qual transporta el material fins al que anomenarem el bloc de trilla.

El bloc de trilla està compost per un cilindre de barres que rep el nom de trilla (6). Aquest element està envoltant en la seva part inferior per una làmina cilíndrica (7). Quan la matèria arriba a aquesta zona, passa entre aquests dos elements, la qual cosa provoca la separació del gra de l'espiga. Una gran part d'aquest gra travessa la còncava per dirigir-se a la part de neteja. L'altra part, la palla, és impulsada per un rotor de paletes cap al sistema de separació.

El bloc de neteja està format principalment per un sacsejador (8), que com el seu nom indica, té la funció de sacsejar la collita, i es realitza així, la separació, també, de la palla i el gra. Aquesta separació es du a terme, degut a que el sacsejador està format per unes reixes, les quals estan en moviment oscil·latori, per les quals cau sols cau el gra quant aquest fa el moviment. En la part baixa del sacsejador trobem un conjunt de garbells (9, 11 i 12), les quals són reixes cada cop amb una mida més petita, per tal que es separi el gra de les impureses, com petites pedres.

A més, per tal que no es malgasti matèria, aquell material que al passar pel trill, no ha sigut separat correctament, trobem un elevador de retorn (13 i 14) que fa que aquest material torni a la trilla i torni a començar el procés.

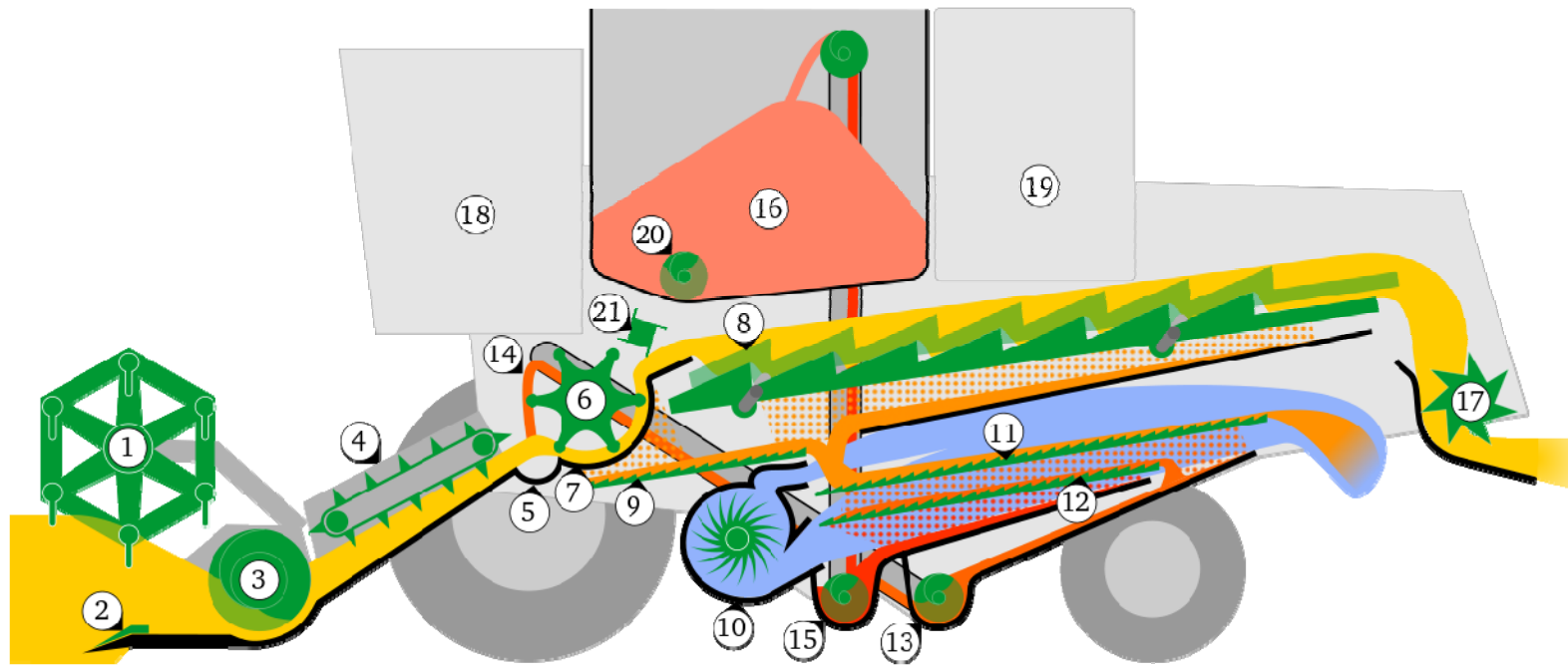


Figura 2: Esquema de la collitadora, amb tots els elements que la conformen.

Finalment, un cop arribat a aquest punt, tenim la matèria que havíem collit en dues parts. Per una part tenim la palla, la qual surt per la part posterior de la collitadora, pel desparramador de palla (17). Per altra banda, tenim el gra, ja lliure d'impureses, el qual és traslladat amb el sense fi elevador de grans (15) fins a la tremuja(16), on és emmagatzemada mentre es du a terme la collita.

Un cop s'acaba la tasca de la collita, sols cal buidar la tremuja per poder emmagatzemar correctament el gra i donar-li la utilitat que l'usuari desitgi.

Així doncs, un cop sabem com funciona una collitadora com la que nosaltres volem millorar, podem entendre millor quina funció té el sistema de sensors que es vol realitzar. Pel que fa als sensors inductius, els quals asseguraran que les rpm d'algunes de les parts mòbils són correctes, si ens fixem amb la figura 3, 6 d'aquests aniran: al trill (I_1), al ventilador (I_2), al picador (I_3), l'elevador classificador (I_4), a l'elevador de reton (I_5) i al sense fi de la màquina (I_6). Els altres dos, controlaran la velocitat del motor general de la màquina (I_7) i els km/h d'avanç d'aquesta (I_8). Al motor general també estan localitzats els sensors de pressió i de temperatura.

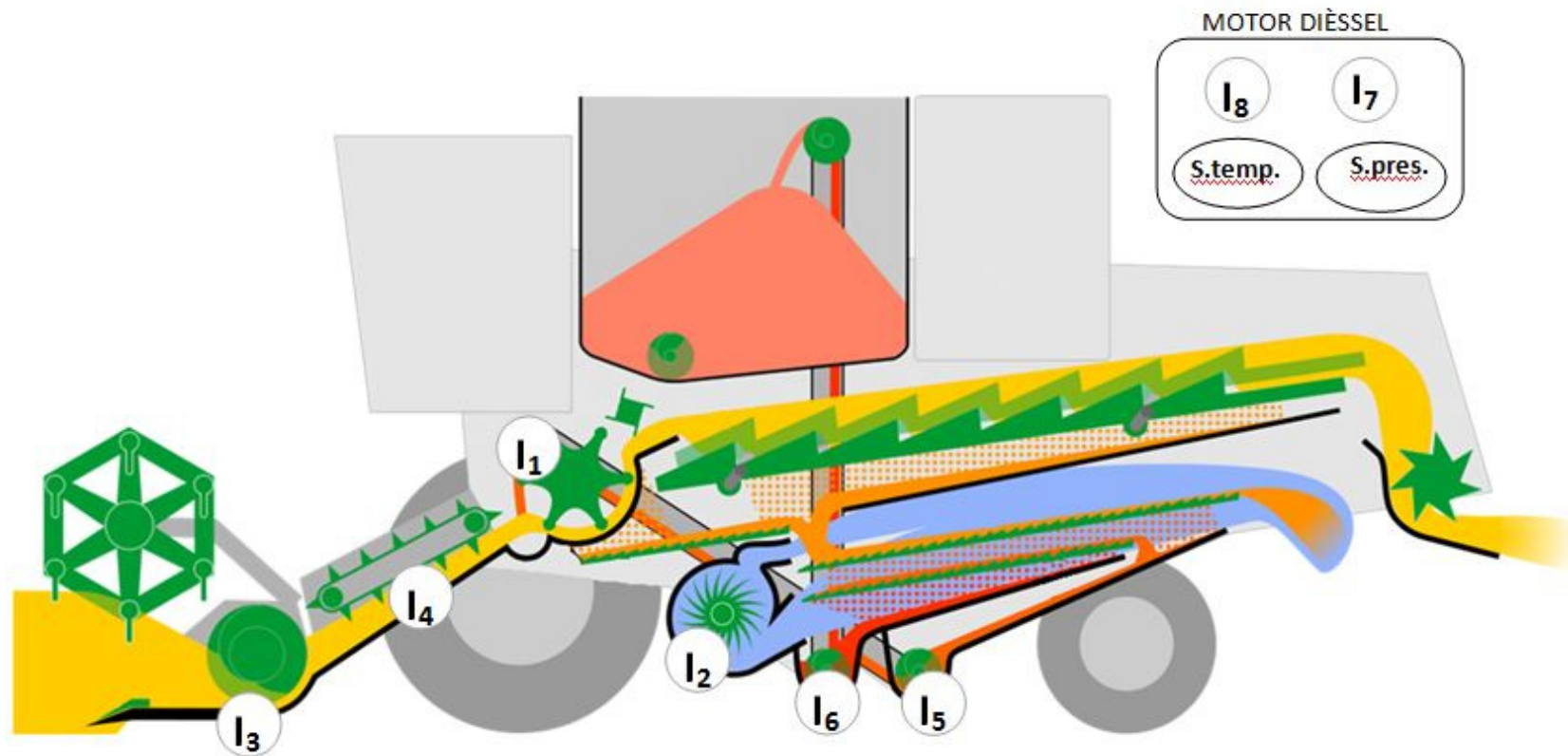


Figura 3: Col·locació dels diferents sensors a la collitadora

Tots aquests sensors estan col·locats de manera estratègica, ja que aquest model de màquina en concret porta 14 corretges de transmissió i 3 cadenes. Així doncs, amb aquesta col·locació dels captadors podem controlar totes les parts mòbils de la màquina, ja que van ubicats a punts terminals. D'aquesta manera, amb la col·locació dels sensors establerta, poden saber quina corretja s'ha trencat o si es bloqueja alguna part mòbil i, per tant, evitar que es trenquin altres peces o evitar un incendi per fricció.

Per altra banda, per tal de detectar les incidències a temps perquè no es produeixin avaries importants, s'instal·larà un sistema d'alarma. Primerament, hi haurà una alarma sonora, un buzzer o altaveus, els quals generaran una senyal a una certa freqüència, quan un dels paràmetres monitoritzats estigui fora de rang.

A més, per tal que aquesta alarma sigui més visible i l'usuari pugui assabentar-se de quin dels paràmetres està fora de rang, utilitzarem un monitor. En aquest monitor es podran veure les dades a temps real dels diferents paràmetres, amb actualització cada 2 segons. A més, quan l'alarma s'activi apareixerà un missatge per pantalla indicant quin paràmetre falla, per exemple "Alarma pressió màx."

Finalment, perquè l'usuari pugui canviar els valors a partir dels quals l'alarma ha d'avisar que un paràmetre està fora de rang, el sistema tindrà connexió de l'Arduino a un PC. Amb aquesta connexió l'usuari podrà canviar els diferents paràmetres del programa.

En resum, com podem veure a la figura 4, tindrem els sensors connectats als diferents punts especificats de la collitadora. Aquests sensors enviaran el senyal a l'Arduino, el qual gestionarà la informació i la mostrarà pel monitor a temps real. A més, també farà sonar l'alarma si fos necessari. Finalment, l'Arduino es podrà connectar directament amb un ordinador, per tal de poder modificar els paràmetres d'alerta d'alarma.

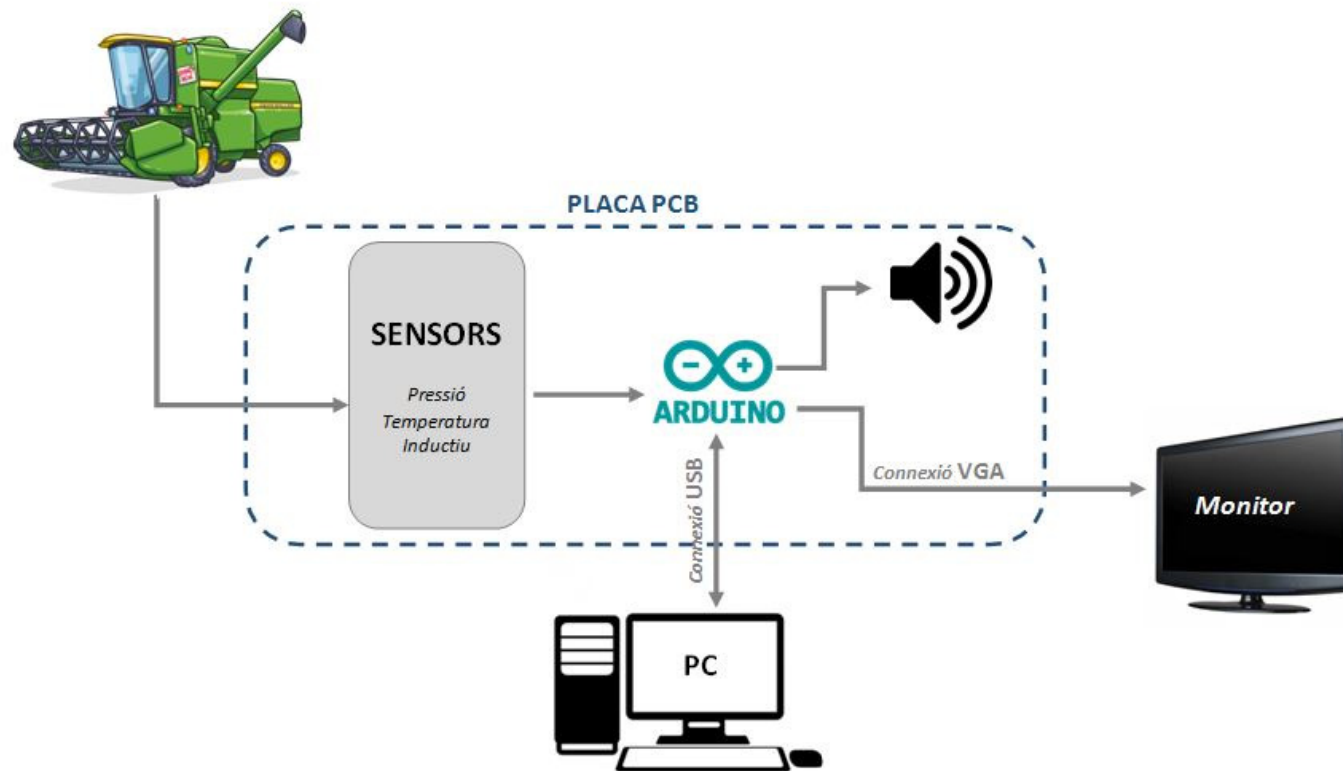


Figura 4: Diagrama de blocs del sistema

4. HARDWARE

Com ja hem comentat, disposem de diversos sensors que ens permeten obtenir els paràmetres que desitgem. Per això utilitzarem sensors inductius, de pressió i de temperatura. A més, també tindrem l'Arduino, el qual serà l'encarregat de gestionar tota la informació i activar les alarmes en el cas que sigui precisos.

4.1 ARDUINO

L'Arduino que s'ha elegit en aquesta ocasió ha sigut l'Arduino DUE^[38], ja que és el que s'ajustava més a les nostres necessitats. El principal motiu d'aquesta elecció és bàsicament la necessitat de tenir mínim 8 entrades d'interruptió, i en aquest model tots els pins digitals són d'interruptió.

Les 8 interrupcions són necessàries per controlar les revolucions dels diferents elements mòbils de la collitadora. Aquestes revolucions es comptaran mitjançant els sensors inductius.

Per tal d'entendre adequadament el muntatge dels circuits i com aquests es relacionen amb l'Arduino és necessari tenir una visió general d'aquest. El model elegit es pot alimentar de 7 a 12V (màx. de 15V). Així doncs, aprofitant que la nostra font d'alimentació general és de 12 V, utilitzarem aquesta tensió d'alimentació per l'Arduino. A més, també hem de tenir en compte que aquest tipus de microcontrolador pot rebre entrades i generar sortides de 0 o 1, on 1 equival a 3,3V.



Figura 5: Arduino DUE

4.2 SENSORS I ADAPTACIÓ DELS CIRUCITS

4.2.1 SENSOR DE TEMPERATURA

Com ja hem mencionat, el sensor de temperatura té l'objectiu de controlar la temperatura del motor dièsel, motor de la collitadora. Aquest sensor ha de tenir un rang de medició de 20°C a 140°C, encara que normalment treballarà a una temperatura de 80-90°C.

A l'hora d'elegir el sensor de temperatura hem estat estudiant diferents opcions. Una d'elles, era usar una PT100^[21], el qual pot anar d'uns -50°C a 450°C, encara que aquest rang depèn molt de la connexió i del tipus de construcció que es faci. Aquest tipus de sensor, a més, te l'avantatge que té un preu baix, el qual ronda els 5 euros.

Per altra banda, tenim l'opció de la Ni200^[18], aquest tipus de sensors són de mot fàcil us, ja que funcionen com una resistència convencional. Aquest sensor té un rang de mesura de -60 a 500 °C i també un preu molt econòmic.

Finalment, també es podria utilitzar un sensor termopar de tipus K, exactament el model 363-0294^[22] de la casa RD PRO. Aquest sensor té un rang de mesura de -60°C a 350 °C, i disposa d'una sensibilitat de 41 µV/°C. Tot i que degut a la seva qualitat té un preu una mica més elevat, de 16,6 €.



Figura 6: Sensor de temperatura, PT100



Figura 7: Sensor de temperatura, NI200



Figura 8. Sensor de temperatura, termopar de tipus K i model 363-0294.

Aquests tres sensors són dels més utilitzats en el món de la indústria, són de baix cost i tenen un funcionament senzill.

Finalment, es va decidir utilitzar la Ni200, perquè com ja hem mencionat, un dels objectius del treball és realitzar aquest sistema amb un baix cost econòmic, i teníem a la nostra disposició aquest tipus de sensor.

4.2.1.1 SENSOR ELEGIT : NI200^[18]

El sensor de temperatura utilitzat és el Ni200, el qual té un funcionament molt simple, es comporta com una resistència variable a la temperatura. En aquest cas, en tractar-se d'un Ni200, significa que a 0° C el valor de la resistència equival a 200 ohms. A partir d'aquí, la corba que segueix aquest sensor és la de l'equació 1.

$$R(t) = Ro \cdot (1 + At + Bt^2 + Ct^4 + Dt^6)Eq. [1]$$

Sent:

$$Ro = 200\Omega$$

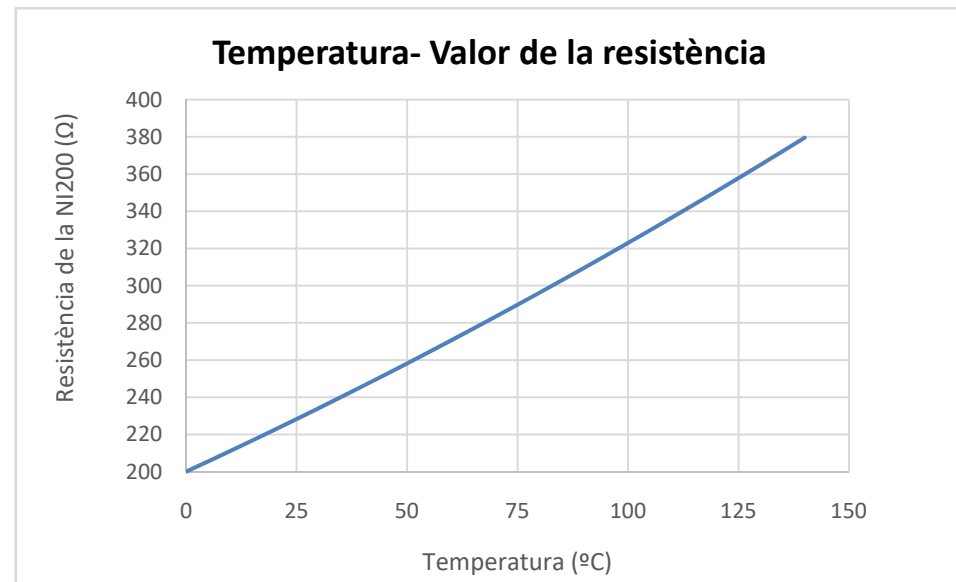
$$A = 5,485 \cdot 10^{-3} \text{ } ^\circ C^{-1}$$

$$B = 6,65 \cdot 10^{-6} \text{ } ^\circ C^{-2}$$

$$C = 2.805 \cdot 10^{-11} \text{ } ^\circ C^{-4}$$

$$D = 2 \cdot 10^{-17} \text{ } ^\circ C^{-6}$$

A partir de la fórmula podem obtenir la recta que relaciona els termes temperatura i resistència, gràfica 1.



Gràfic 1: Relació de temperatura-resistència de la NI200

4.2.1.2 ADAPTACIÓ DEL SENSOR TEMPERATURA

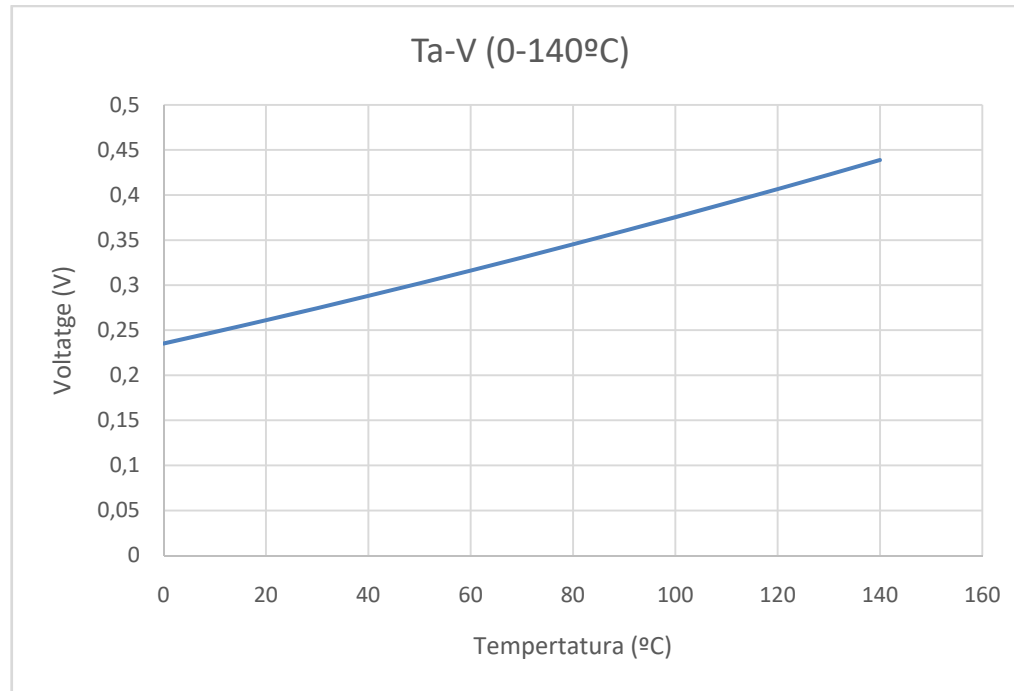
Per tal de poder obtenir la informació que desitgem del sensor ens cal dissenyar un circuit que ens permeti saber la temperatura amb la informació que arriba a l'Arduino. Com el fet que variï la capacitat de la resistència no ens serveix per poder llegir-ho amb l'Arduino el que s'ha fet és convertir aquesta variació de resistència amb una variació de voltatge.

Per poder fer-ho, s'ha dissenyat el circuit que podem observar en la figura 9. Aquest circuit, el qual està alimentat a 12V (V_{CC}), no és més que una divisió de tensió, amb una resistència $R_1=10K$. Per tant, segons el valor de resistència que adquireixi el sensor, la diferència de voltatge (V_a-V_b) en ell serà major o menor.

És a dir, si estem a 0 °C, el valor de la seva resistència és 200 Ohms i per tant la diferència de tensió serà:

$$V(t) = R(t) \cdot I_T \text{ Eq. [2]}$$
$$\rightarrow I_T = \frac{V_{CC}}{\sum R} = \frac{V_{CC}}{R(t) + R_1} = \frac{12}{200 + 10000} = 117,64 \text{ mA} \rightarrow V(t) = 200 \cdot 117,64 \cdot 10^{-3} = \mathbf{0,2352 \text{ V}}$$

En canvi, si per contrari la temperatura és de 140 °C, la resistència passarà a valdre uns 380 ohms i en consegüent la diferència de tensió de 0,438V. Per tant, tenim un rang de tensions de 0.204 V.



Gràfic 2: Relació de Temperatura-Voltatge entrada amplificador

Com podem observar, el valor que obtenim amb aquesta tensió és de dimensions petites, de valor de mV. Si volem que el nostre sistema tingui una sensibilitat de 0,5 °C com ja hem mencionat, amb aquest mínim interval de dades no ens serà possible. Per això, s'ha hagut de realitzar una amplificació del senyal.

L'amplificador utilitzat és el UA741^[29], ja que és dels més comercials. Per saber quin es màxim guany possible, el que permetre una major sensibilitat, s'ha analitzat el circuit amplificador de la figura 9. Les equacions que descriuen el circuit són:

$$Eq. [3] \rightarrow \frac{V_a - V_c}{R_2} = \frac{V_c - 0}{R_4} \rightarrow R_4 (V_a - V_c) - R_3 \cdot V_c = 0$$

$$Eq. [4] \rightarrow \frac{V_b - V_d}{R_3} = \frac{V_d - V_o}{R_5} \rightarrow R_5 (V_b - V_d) - R_3 (V_d - V_o) = 0$$

A més, s'ha de tenir en compte que la tensió a les dos entrades de l'amplificador han de ser iguals. Per tant: $V_c = V_d$. A més, també tenim que $R_2 = R_3$ i que: $R_4 = R_5$.

Si igualem les dos equacions tenim:

$$R_4 (V_a - V_c) - R_3 \cdot V_c = R_5 (V_b - V_d) - R_3 (V_d - V_o) \rightarrow R_5 \cdot V_b + R_2 \cdot V_o - R_5 \cdot V_a = 0 \rightarrow R_5 \cdot (V_b - V_a) = -R_2 \cdot V_o \rightarrow$$

$$\rightarrow V_o = \frac{R_5}{R_2} (V_a - V_b) Eq. [5]; \quad \frac{R_5}{R_2} = \text{Guany} Eq. [6]$$

Ara que ja tenim la relació de resistències per obtenir el guany, hem de calcular quin és el guany màxim que podem tenir, per tal s'aprofitarà al màxim el rang que l'Arduino ens permet: 3,3V.

$$\text{Així doncs: } \begin{cases} Rang_{del'Arduino} = 3,3 V \\ Rang_{desortidadel sensor} = 0,204 V \end{cases} \rightarrow Guany_{màxim} = \frac{3,3}{0,204} = 16,17$$

Però, encara que el guany màxim pugui ser 16, establim un guany de 15. Això, és per prevenir i deixar un marge perquè l'Arduino pugui fer una correcta lectura.

$$\text{Establim que } R_2 = R_3 = 10K. \text{ Per tant: } Guany = \frac{R_5}{R_2} = \frac{R_5}{10K} = 15 \rightarrow R_5 = 150K$$

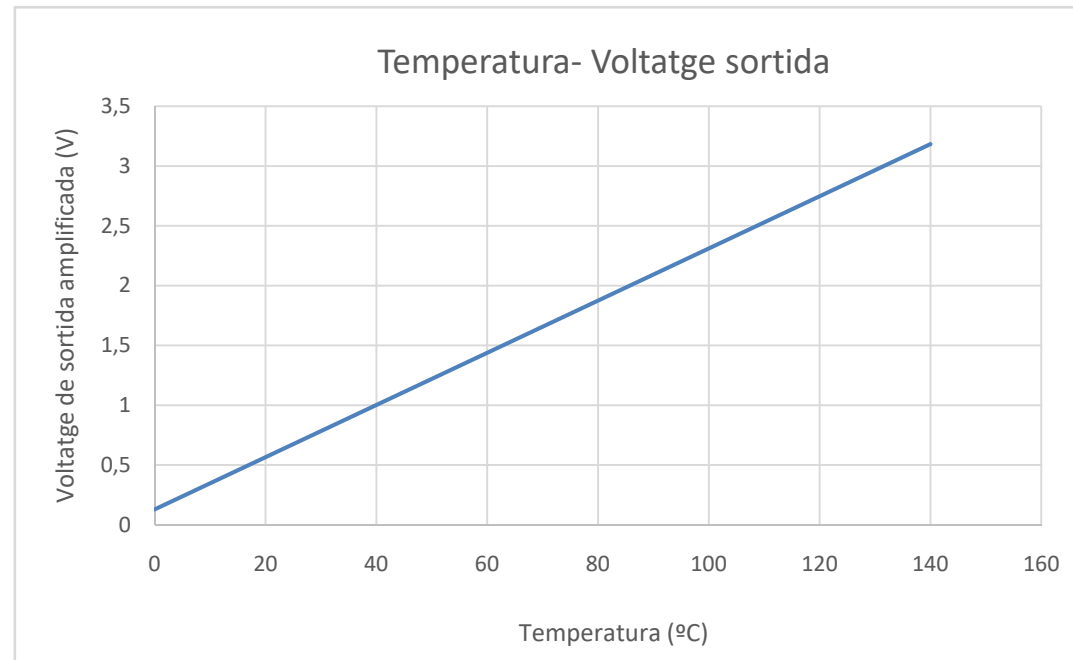
Així doncs un cop tenim el guany amb el qual treballarà l'amplificador tenim que el rang de sortida d'aquest serà de 3.5V a 6.6V.

Com ja hem mencionat, aquest rang no és adequat per ser llegit amb l'Arduino. Per tant, és necessari baixar el senyal per poder gestionar-la i llegir-la correctament amb l'Arduino, ja que com hem mencionat, aquest sols pot llegir correctament valors de, com a màxim, 3,4V.

Una de les solucions a primera vista més simples és la incorporació d'una font negativa al circuit amplificador, amb l'inconvenient que en aquest cas no disposem d'una font negativa principal. Així doncs, per aconseguir-ho ho farem mitjançant la font de 12V.

El MAX 233^[32], és utilitzat normalment en comunicacions d'interfície, entre un sistema de processament de la informació (PC) i les persones, o un altre sistema de processament de la informació. Però, en alguns casos, també és utilitzat per aconseguir fonts de tensió negatives. Així doncs, aquest component és idoni per poder fer aquesta conversió d'una font positiva a una negativa, ja que ens ho genera ell automàticament i de forma molt senzilla. En introduir-li una font de, en aquest cas +5V, la sortida d'aquest xip serà de $\pm 10V$. Com la font que necessitem en aquest cas és de -3,4V, hem hagut d'introduir una altra divisió de tensió, el qual ens permetrà reduir els -10 V de sortida als -3,4V que nosaltres ens interessa tenir.

Després de tot aquest procés de tractament de senyal del NI200, el que obtenim és una relació de temperatura amb voltatge, el qual és llegit per Arduino, que segueix la corba de la gràfica 3.



Gràfic 3: Relació Temperatura- Voltatge que arriba al pin de l'Arduino.

Finalment, cal mencionar que la sortida de voltatge que apareix en el gràfic superior serà la que llegirà l'Arduino. Com aquest valor és analògic, ja que pot comprendre valors entre 0.1V i 3,2.V, hem de connectar-lo a un pin amb aquestes característiques, i per tant s'ha connectat al pin A2.

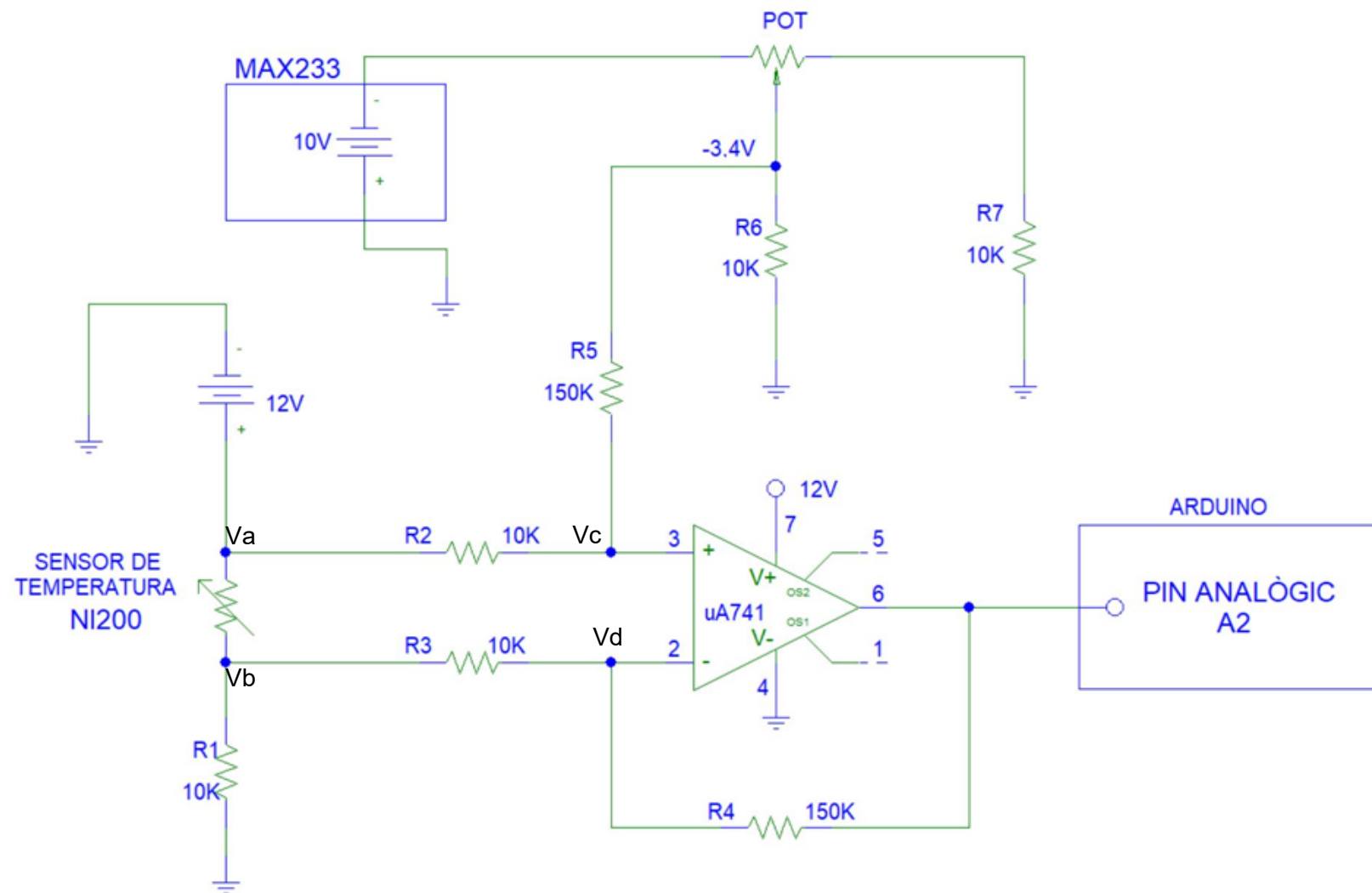


Figura 9: Circuit d'adaptació del sensor de temperatura NI200

4.2.2 SENSOR DE PRESSIÓ

El sensor de pressió té l'objectiu de controlar la pressió del motor dièsel. Aquesta ha de poder ser mesurable en un rang d'entre 0 a 8 Bars, encara que quan el sistema funcioni adequadament, aquest treballarà entre 4 i 6 bars.

Per poder dur a terme aquesta feina podem optar per diferents tipus de sensor. Per una banda, podríem elegir el sensor de pressió WIKA-10A^[24]. Aquest sensor té varies funcions i modes de connexió, els quals fan que aquest funcioni de manera diferent. En aquest cas, es connectaria amb el mode de 2 fils. Si es connecta en sensor d'aquesta manera, aquests funciona com una font de corrent, amb corrents variants de 4 mA a 20 mA, corresponent a una pressió de 0 bars a 10 bars respectivament. Aquest sensor té un preu aproximat d'uns 135€.

Per una altra banda, també podríem elegir el DIN43650c^[23] Industrial, de la casa GemsSensor. Aquest sensor, igual que el WIKA-10A té varies funcions i modes de connexió, els quals fan que aquest funcioni de manera diferent. En aquest cas, es connectaria amb el mode de 2 fils. Si es connecta el sensor d'aquesta manera, aquest funciona com una font de corrent, amb corrents variants de 4 mA a 20 mA, corresponent a una pressió de 0 bars a 10 bars respectivament. Aquest sensor pot treballar fins a temperatures de 125°C, per sobre d'aquesta no s'assegura el correcte funcionament d'aquest. Pel que va a la tensió d'alimentació, el ventall es molt obert, de 10 V a 30 V de corrent continua. Aquest sensor costaria al voltant d'uns 122,70€.

Finalment, una altra opció seria un sensor de pressió de la casa IFM, concretament és el model PT5415^[25]. Aquest sensor té tan sols un tipus de connexió de sortida de 2 fils. Aquesta sortida és analògica de 4 a 20 mA, que correspon a una pressió de 0 a 6 bars respectivament. Aquest sensor pot treballar fins a temperatures de 90°C, per sobre d'aquesta no s'assegura el correcte funcionament d'aquest. Pel que va a la tensió d'alimentació el ventall és molt obert (de 8.6 V a 36 V de corrent continua). Aquest sensor seria el més econòmic dels tres, ja que té un preu de 102,10 €.



Figura 12: Sensor de pressió de la casa IFM, model PT5415.



Figura 11: Sensor de pressió WIKA-A10



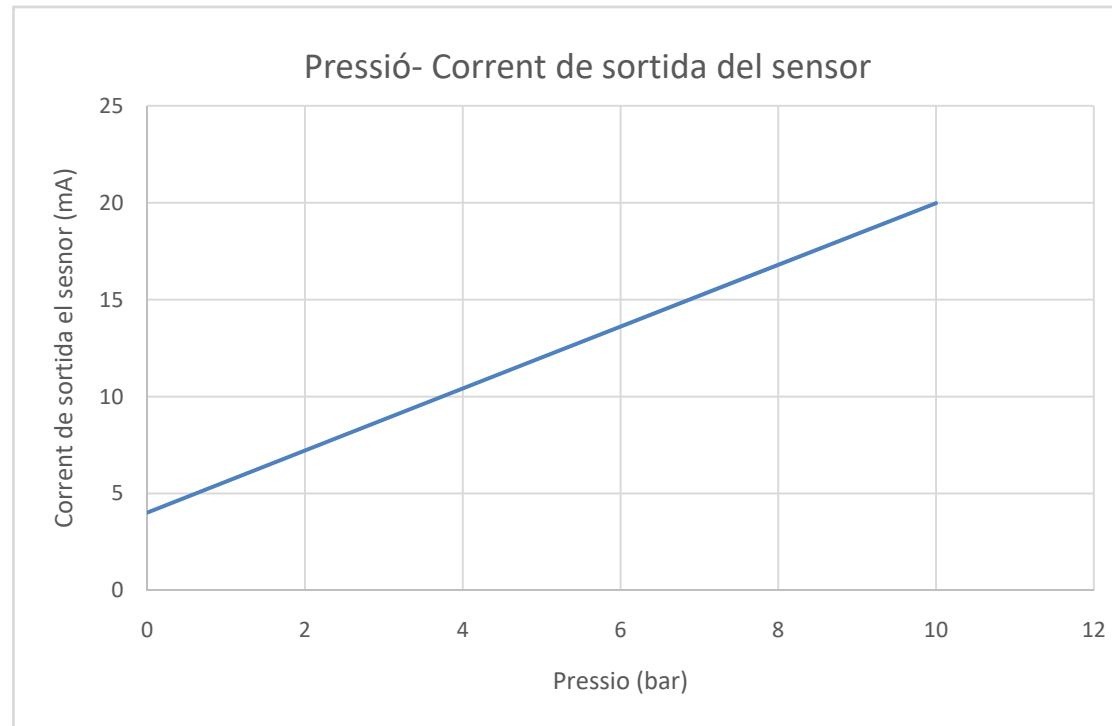
Figura 10. Sensor de pressió, model DIN43650c, de la casa GemsSensor

Així doncs, estudiades les possibilitats s'ha decidit utilitzar el WIKA-A10, ja que complia amb el rang de mesura que el nostre sistema requeria i, a més, té una connexió simple.

4.2.2.1 SENSOR ELEGIT: WIKA-10A^[16]

El sensor de pressió elegit és un WIKA-10A. Si es connecta el sensor amb 2 fils, aquest funciona com una font de corrent, amb corrents variants de 4 mA a 20 mA, corresponent a una pressió de 0 bars a 10 bars respectivament.

És a dir, el sensor ens mesurarà una pressió del motor entre 0 i 10 bars, i ens proporcionarà un corrent de 4-20 mA, com podem veure en la gràfica 4. A més, cal tenir en compte que encara que el sensor tingui aquest ventall de mesura, normalment, el nostre motor funcionarà entre 4 i 6 bars, i el sistema avisarà quan aquesta pressió sigui inferior a 3 bars.



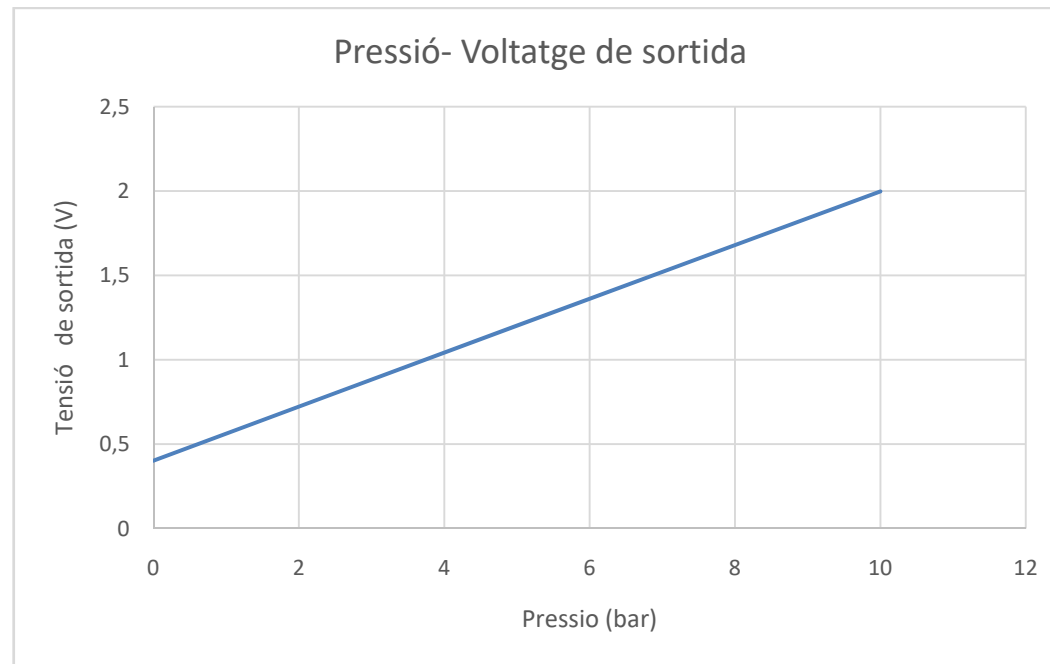
Gràfic 4: Relació de pressió i corrent de sortida del sensor WKA-A10.

4.2.2.1.2 ADAPTACIÓ DEL SENSOR DE PRESSIÓ

Com ja hem mencionat, el sensor de pressió elegit funciona com a una font corrent. Com l'Arduino sols pot llegir entrades de tensió, ens cal transformar aquest corrent de sortida del sensor a tensió. Per fer-ho, simplement farem que aquest corrent, el qual ens dóna la informació, passi per una resistència, com mostra la figura 13. D'aquesta manera, el que farem serà agafar la caiguda de tensió en aquesta resistència com a informació per extreure la pressió.

La residència R1, de la figura 13, té un valor de 100 ohms. Per tant, seguint la llei de Ohm:

$$Eq. [7] \quad V = R \cdot I \rightarrow \begin{cases} Si I = 4 \text{ mA} \rightarrow V = 100 \cdot 0.004 = 0.4 \text{ V} \\ Si I = 20 \text{ mA} \rightarrow V = 100 \cdot 0.02 = 2 \text{ V} \end{cases}$$



Gràfic 5: Relació pressió- tensió en R1.

Per tant, en aquest cas, V_a - V_b anirà de 0.4 a 2 V. Un rang de 1.6V.

Igualment, com hem fet amb el sensor de temperatura, per tal d'aprofitar al màxim el rang de tensió que l'Arduino pot llegir,

amplificarem aquest senyal que hem obtingut. El circuit amplificador serà el mateix que en el cas del sensor de temperatura. Per tant la relació de tensions serà la mateixa.

$$V_o = \frac{R_6}{R_3} (V_a - V_b) \text{ Eq. [5]}; \quad \frac{R_6}{R_3} = \text{Guany} \text{ Eq. [6]}$$

Ara que ja tenim la relació de resistències per obtenir el guany, hem de calcular quin és el guany màxim que podem tenir, per tal d'aprofitar al màxim el rang que l'Arduino ens permet 3,3V.

$$\text{Així doncs: } \begin{cases} \text{Rang del Arduino} = 3,3 \text{ V} \\ \text{Rang de sortida del sensor} = 1,6 \text{ V} \end{cases} \rightarrow \text{Guany màxim} = \frac{3,3}{1,6} = 2,06$$

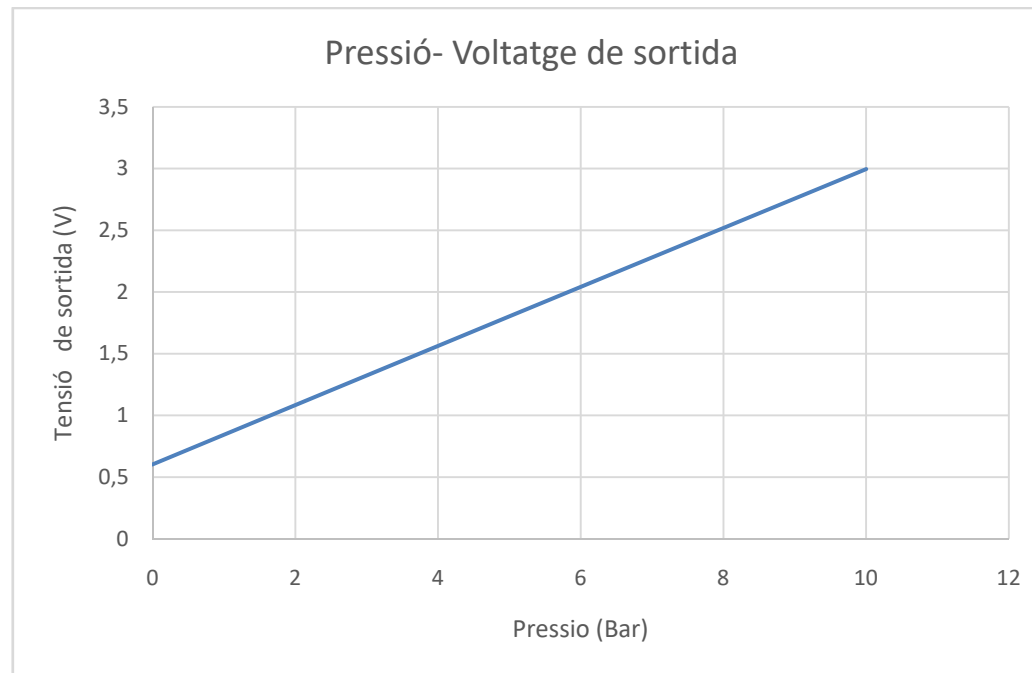
Però, encara que el guany màxim pugui ser 2, establim un guany de 1,5. Això és per prevenir i deixar un marge entre el màxim i el mínim que l'Arduino pot llegir.

$$\text{Establim que } R_2 = R_3 = 100\text{K. Per tant: } \text{Guany} = \frac{R_5}{R_2} = \frac{R_5}{10\text{K}} = 1,5 \rightarrow \mathbf{R_5 = 150\text{K}}$$

En aquest cas, establim que $R_2 = R_3 = 100\text{K}$, degut a que ens interessa que vagi el mínim de tensió procedent de la font de tensió del sensor. Per tant, si aquesta resistència és més gran, que en el cas anterior, per exemple, aconseguirem aquest fet.

Així doncs, un cop tenim el guany amb el que treballarà l'amplificador tenim que el rang de sortida d'aquest serà de 0.6V a 3V.

En el cas del sensor de pressió, el circuit necessari és menys complexa que en el cas del sensor de pressió, ja que no tenim que baixar la tensió de referència, i queda com es mostra a la figura 13.



Gràfic 6: Relació pressió- voltatge que arriba al pin A3 de l'Arduino.

Aquest sensor anirà connectat al pin A3 de l'Arduino Due, ja que per captar bé la informació el pin ha de ser analògic.

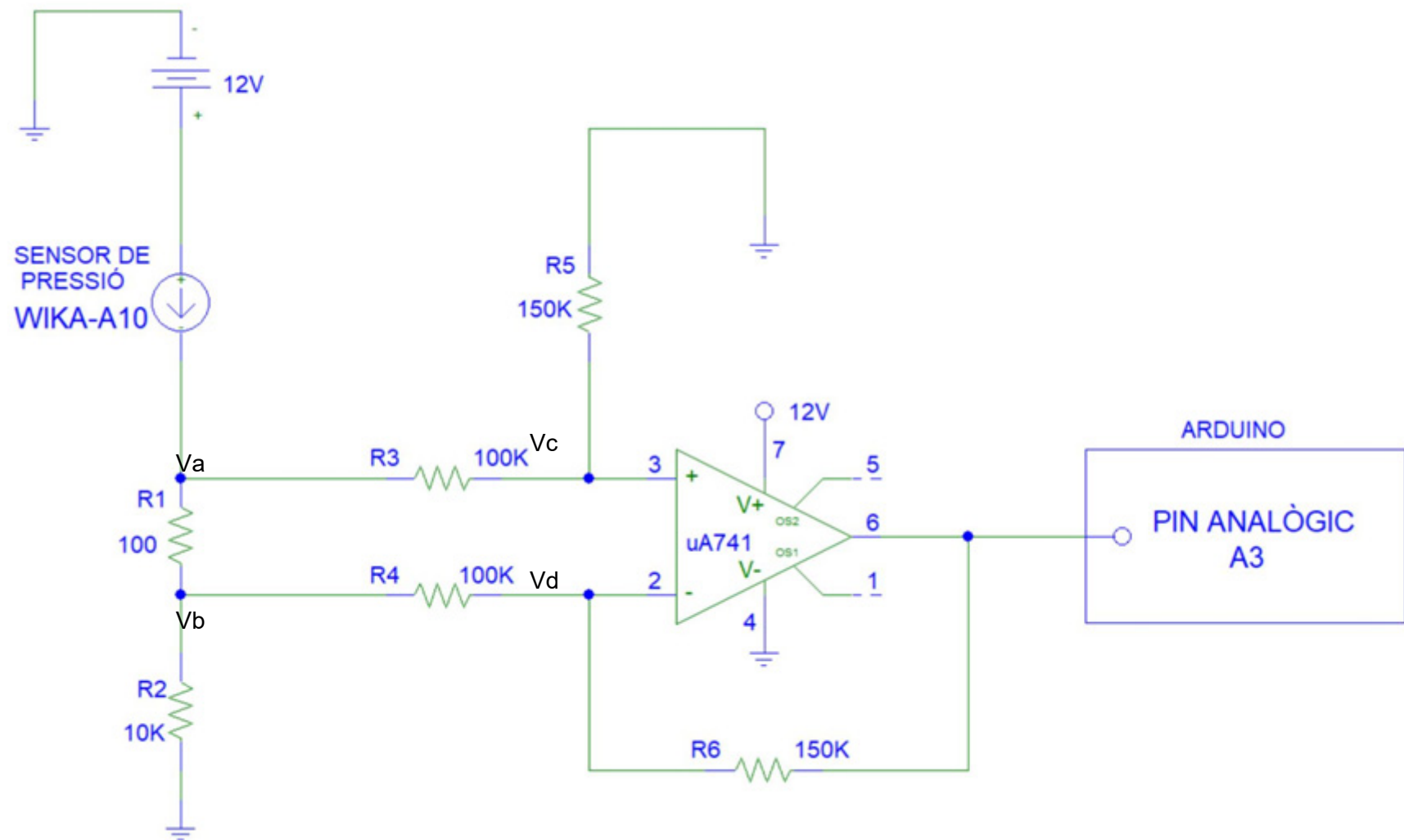


Figura 13: Circuit d'Adaptació del sensor de pressió.

4.2.3 SENSORS INDUCITUS

Finalment, l'últim sensor del nostre sistema és el sensor inductiu. Aquest sensor, com ja hem comentat, tindrà la funció de controlar la velocitat dels diferents motors. Per això, com tenim 8 motors a controlar, haurem de disposar de 8 sensors inductius iguals, però independents l'un de l'altre.

Els sensors inductius funcionen per flancs de pujada. Això significa que cada vegada que el sensor detecta un metall a certa distància, aquest tanca el circuit generant un senyal de certa tensió.

Un sensor inductiu és del model IME12-04BPSZCOS (1040764)^[27]. Com gairebé tots els sensors inductius, aquest funciona mitjançant impulsos, el que significa que quan aquest detecti un metall a menys de 4mm ens proporcionarà una tensió. Per contra, si no detecta res, no ens arribarà tensió. A més, aquest sensor té un preu no molt elevat, d'uns 40€.

Una altra opció seria el sensor inductiu, NBB5-18GM50-E2-V1 de la casa Pepperl+fuchs^[26]. Aquest sensor té una tensió de treball de 10 V a 30 V, de corrent continua. És del tipus normalment obert (no proporciona tensió mentre no en detecti), la sortida és PNP i la distància de detecció és de 5mm. La sortida és de 4 fils però un d'ells no es fa servir. Un és positiu, un negatiu i l'altre és el senyal. Aquest sensor té un preu de 68.97€.

Finalment, una altra opció seria el sensor inductiu model BES M08EC-PSC20B-S49G^[28], de la casa BALLUFF. Aquest sensor té una tensió de treball de 10 V a 30 V, de corrent continua. És del tipus normalment obert (no proporciona tensió mentre no en detecti), la sortida es PNP i la distància de detecció és de 2mm com a màxim. La sortida, igual que el sensor anterior NBB5-18GM50-E2-V1 és de 4 fils.

Tenint en compte les característiques que ha de tenir el nostre sensor, s'ha decidit utilitzar el sensor model IME12-04BPSZCOS (1040764) , ja que és de fàcil muntatge, i a més té una distància de detecció major que els altres dos. L'elegit pot treballar a fins a 4 mm, mentre que els altres dos, sols a 2mm. A més, també, aquest és un dels més barats, característica a tenir en compte, ja que haurem d'implementar-ne 8 d'iguals.



Figura 16: Sensor inductiu de la casa model BES M08EC-PSC20B-S49G, de la casa BALLUFF



Figura 15: Sensor inductiu de la casa Pepperl+fuchs, i model NBB5-18GM5-E2-V1.



Figura 14: Sensor inductiu, model IME12-04BPSZCOS

4.2.3.1 SENSOR INDUCTIU ELEGIT: IME12-04BPSZCOS ^[17]

Encara que tenim 8 sensors inductius, tots funcionen de la mateixa manera, ja que tots ens llegeixen les revolucions de gir dels diversos motors.

El sensor inductiu és el model IME12-04BPSZCOS (1040764), que al detectar un metall a menys de 4mm ens proporcionarà una tensió. Al contrari, si no detecta, no ens arribarà tensió.

A més tenim l'avantatge que aquest sensor es pot connectar de 8 a 30 V, per tant es pot connectar als 12 volts, alimentació general de la collitadora.

4.2.3.2 ADAPTACIÓ DEL SENSOR

Pel que fa al muntatge d'aquest sensor, l'alimentarem a 12 V. Com la tensió de sortida que ell ens proporcionarà serà la mateixa a la qual està alimentat, funciona com un interruptor, i serà de 12V. Com ja hem mencionat repetides vegades, aquest valor no pot ser llegit per l'Arduino, ja que aquest opera en un rang de 0-3.3V. Per tant, per tal que quan el sensor ens proporcioni aquest senyal de 12V i a l'Arduino li arribi un senyal proper a 3V, hem incorporat un circuit que, bàsicament, està format per un transistor, BC108. Aquest circuit és el de la figura 17.

El transistor BC108^[30] és de tipus NPN. Per tant, el que volem aconseguir amb aquest transistor és que quan el sensor no detecti, aquest estigui en mode de tall, la qual cosa voldrà dir que no deixarà passar a corrent de l'emissor al col·lector, i, per tant, no arribarà tensió al nostre Arduino. Si per contra, el transistor està en mode de saturació, el corrent passarà de l'emissor al col·lector i per tant la informació que ens proporciona el sensor arribarà a l'Arduino. D'aquesta forma, i aplicant l'explica't, quan el sensor ens doni un senyal de 12V, a l'Arduino ens arribarà un voltatge, de 3V.

Per assegurar que aquest transistor es adequat analitzarem el circuit. Si tenim:

$$\begin{cases} R_2 = R_{col\cdot lector} = 10K \\ R_3 = R_{emissor} = 3K \end{cases}$$

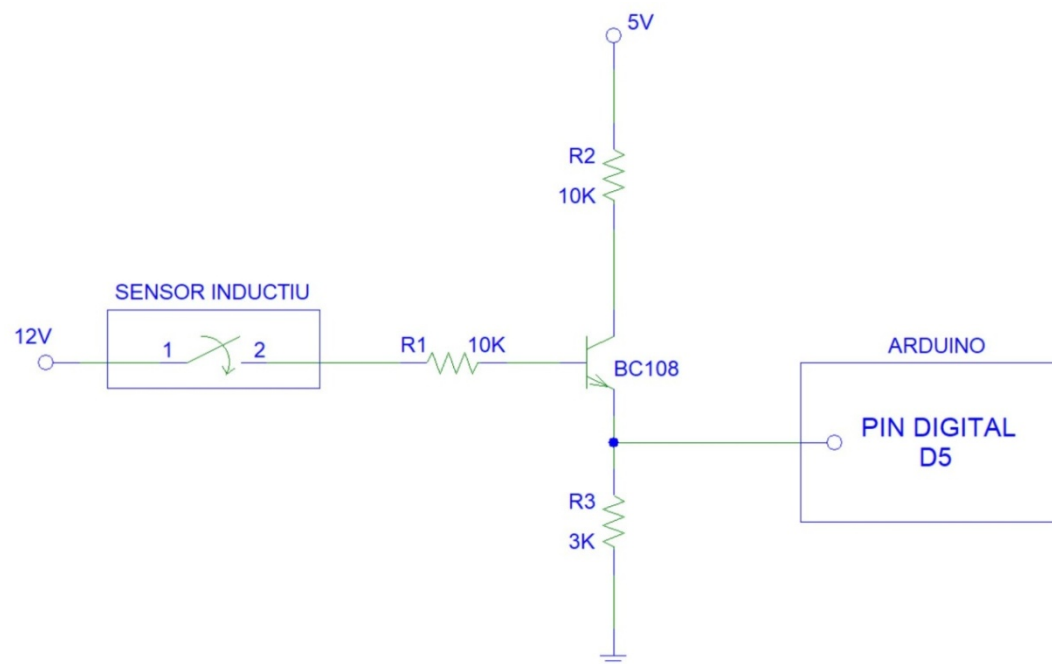


Figura 17: Circuit d'adaptació dels sensors inductius.

Per poder aconseguir els 3V per l'Arduino, s'ha de complir l'expressió:

$$R_e \cdot (I_b + I_c) = 3V \rightarrow 3k \cdot (I_b + I_c) = 3V \rightarrow I_b + I_c = 1mA \text{ Eq. [8]}$$

Aleshores, analitzem la malla de sortida, tenint en compte que si el transistor està saturat $V_{ce}=0,2$ V.

$$\begin{aligned} \text{Eq [9]} \quad V_{cc} - V_{ce} &= R_c \cdot I_c + R_e \cdot (I_b + I_c) \rightarrow 5 - 0.2 = (10 \cdot 10^3) \cdot I_c + (3 \cdot 10^3) \cdot (1 \cdot 10^{-3}) \rightarrow I_c = \frac{5 - 0.2 - (3 \cdot 10^3) \cdot (1 \cdot 10^{-3})}{(10 \cdot 10^3)} \rightarrow \\ &\rightarrow I_c = 0,18 \text{ mA} \end{aligned}$$

Ara que ja sabem el valor de I_c , podem extreure el valor de I_b , a partir de l'equació 8.

$$I_b + I_c = 1mA \rightarrow I_b = 1 \text{ mA} - I_c = 1mA - 0,18 \text{ mA} \rightarrow I_b = 0,82 \text{ mA}$$

Analitzem la malla d'entrada. ($V_{be}=0,7$)

$$\begin{aligned} \text{Eq [10]} \quad V_{BB} - V_{be} &= R_b \cdot I_b + R_e \cdot (I_b + I_c) \rightarrow 12 - 0.7 = R_b \cdot (0,82 \cdot 10^{-3}) + (3 \cdot 10^3) \cdot (1 \cdot 10^{-3}) \rightarrow R_b = \frac{12 - 0.7 - (3 \cdot 10^3) \cdot (1 \cdot 10^{-3})}{(0,82 \cdot 10^{-3})} \\ R_b &= 10121,95 \rightarrow R_b = 10K \end{aligned}$$

Per tant, si establim un valor de $R_b=10K$, el transistor del tipus BC108 és adequat.

El valor que arriba a l'Arduino és també digital, ja que com funciona amb impulsos, sols ens és necessari saber quan aquest ens dóna informació. Per tant, en aquest cas, aquest sensor anirà connectat a un pin digital.

Aquest procés, l'hem repetit igualment en els altres 7 sensors, fent així que la connexió dels 8 sensors inductius es distribueixi entre el pin 4 (motor 1) fins el 11 (motor 8).

4.3 SISTEMA DE ALARMA

El sistema de sensorat, tindrà tan alarma visual com sonora, per tal que l'usuari s'assabenti quan un dels paràmetres no es l'adequat.

4.3.1 ALARMA VISUAL

Per implementar l'alarma visual ho hem fet mitjançant un monitor. Per connectar aquest monitor amb el nostre Arduino teníem dues opcions, mitjançant un cable RCA (Radio Corporation of America), o mitjançant un cable VGA (Video Graphics Array). A l'hora d'escollir una de les dues connexions ens vàrem basar en la programació i muntatge que requeria cadascuna.

Per una banda, la connexió RCA és la més comú i senzilla de realitzar, ja que només requeria de 2 resistències per tal de transmetre el senyal d'imatge, una altre per la de sincronització i, finalment, una connexió per transmetre el senyal de terra de l'Arduino. Pel que fa a la programació fa servir la seva pròpia llibreria *ArduitoTvOut*, la qual només està disponible per a l'Arduino Uno, ja que la nostra Arduino no té els pins de funció que utilitza la llibreria. Per tant aquest tipus de connexió queda descartada.

En canvi la connexió VGA requeria el seu propi connector DB15, del qual necessitàvem 5 pins per transmetre senyal i de 5 pins connectats entre si per fer la connexió a GND. Respecte a la programació fa servir la seva pròpia llibreria, *ArduinoVGA*, la qual està adaptada per a la nostra placa Arduino Due, amb el nom de *ArduinoDueVGA*, aquesta llibreria no és pròpia dels desenvolupadors d'Arduino, sinó que està adaptada per una persona externa. Per tant vam escollir aquesta connexió.

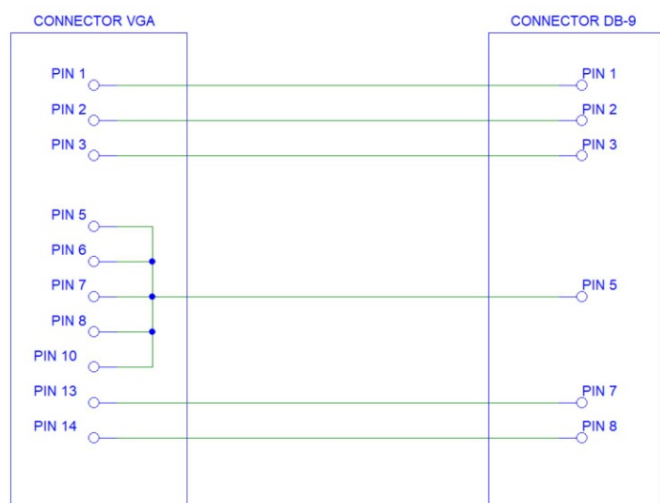
4.3.1.1 VGA

La connexió escollida finalment és la VGA, aquest tipus de connexió requereix de 5 pins per transmetre senyal i uns altres 5 units per transmetre el senyal de terra, com es mostra a la figura 18. Aquest cable tenen una funció pròpia, els pins 1,2,3, transmeten el senyal de colors Vermell, Verd i Blau, mentre que els pins 13 i 14 transmeten el senyal de Hsync i Vsync (sincronització horitzontal i sincronització vertical de la imatge).

En aquest cas no vàrem tallar un cable VGA que tinguéssim, ja que com no faríem servir tots els pins, vam pensar que era molt millor fer nosaltres el cable, amb un connector DB15 mascle i una mànega amb 6 cables.

PIN HD15	FUNCIÓ	COLOR
1	Red	VERMELL
2	Green	VERD
3	Blue	BLAU
5 + 6 + 7 + 8 + 10	GND	BLANC
13	HSync	MARRÓ
14	VSync	GROC

Taula 1: Pins i funcions del VGA



Per tal de no confondre els pins amb el cablejat, vam utilitzar els colors propis de cada pin, menys en el cas del pin de GND, al qual li vam canviar el color a blanc.

Pel que fa a la connexió a l'Arduino, com la nostra senyal al monitor és monocromàtica (pantalla en negre i els missatges en blanc o al revés), a la llibreria ens indica com realitzar les connexions. Els cables 1,2,3 (vermell, verd, blau) aniran connectats a una resistència de 100Ω, i aniran al pin central inferior del SPI, al pin MOSI.

Figura 18: Connexió del DB-15 amb el DB-9

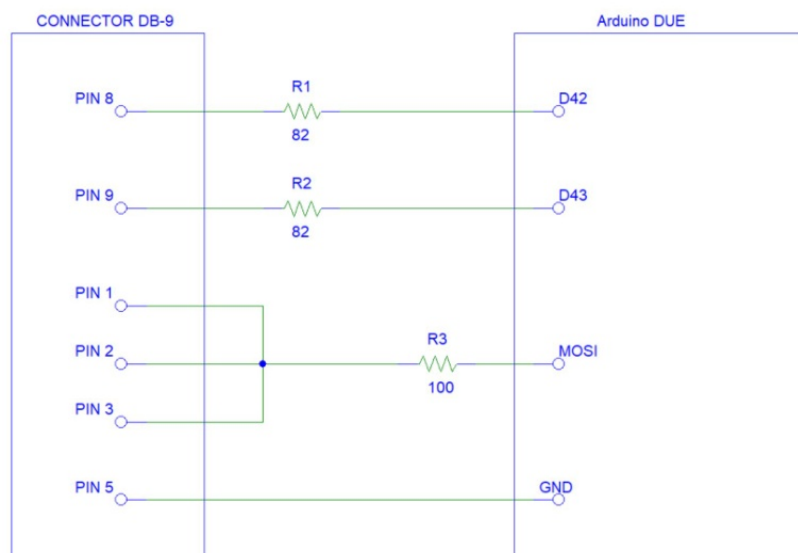


Figura 19: Connexió del DB-9 amb l'Arduino DUE

El cable blanc del senyal de GND, anirà connectada directament al terra de la nostra placa. Els cables marró i groc, referents a la sincronització horitzontal i vertical de la pantalla aniran connectats als pins digitals 42 i 43 a través d'una resistència de 82 Ω.

Per tal de no connectar els cables a una regleta per a PCB, el que podia suposar una equivocació per l'usuari, vam decidir que a l'altra punta de la mànega de cables, hi anés un connector DB 9 mascle, que aniria connectat a un connector DB 9 femella situat a la placa PCB, d'aquesta manera, l'usuari no tindria forma d'equivocar-se en connectar la pantalla.

4.3.2 ALARMA SONORA

Per poder realitzar l'alarma sonora, podríem fer-ho, principalment, de dues maneres. La primera i la més senzilla és utilitzant un buzzer, un "zumbador", el qual generarà soroll a una certa freqüència. L'altra opció seria fer-ho amb un altaveu. Però, com estem intentant realitzar un sistema econòmic, i el buzzer, a més, ocuparà menys espai, ja que pot ser soldar directament a la placa, hem decidit utilitzar aquesta opció.

Així doncs, l'alarma sonora es generarà amb un buzzer. Aquest és del model 32S4120LF^[33], el qual pot treballar de 6 a 16V, i per tant, si el fem funcionar a 12 V, serà quan aquest provoqui una ona sonora més forta, fet important tenint en compte l'entorn en el qual treballarà el nostre sistema.

L'Arduino serà qui donarà l'ordre de quan aquest buzzer ha de generar soroll o no. El problema que sigui l'Arduino qui doni el senyal és que aquets sols pot proporcionar un voltatge de 3.3V, voltatge que no podrà fer funcionar adequadament el component. Com a més voltatge s'alimenti el buzzer, més fort sonarà, cosa que ens interessa, ja que el nostre sistema treballarà en un entorn amb molt de soroll. Així doncs, hem dissenyat un circuit molt semblant a l'utilitza't amb els sensors inductius, però en sentit contrari, de manera que el buzzer estigui alimentat a 12V, tensió màxima a la qual disposem.

Es a dir, en aquest cas, també utilitzarem un transistor NPN, en aquest cas el model MC140^[31]. Quan l'Arduino doni l'ordre de que el buzzer soni, el NPN passarà d'estar en mode de tall a saturació i, per tant, els 12V amb els quals estarà directament alimentat el buzzer faran sonar aquest. Per contra, si l'Arduino no dona senyal, el circuit no es mantindrà en mode de tall i, per tant, no sonarà l'alarma.

El valor de la resistència del buzzer (R_c) és de 50K. A més, també sabem que la β del transistor és de 40. Per calcular el valor de R_b , hem de saber que:

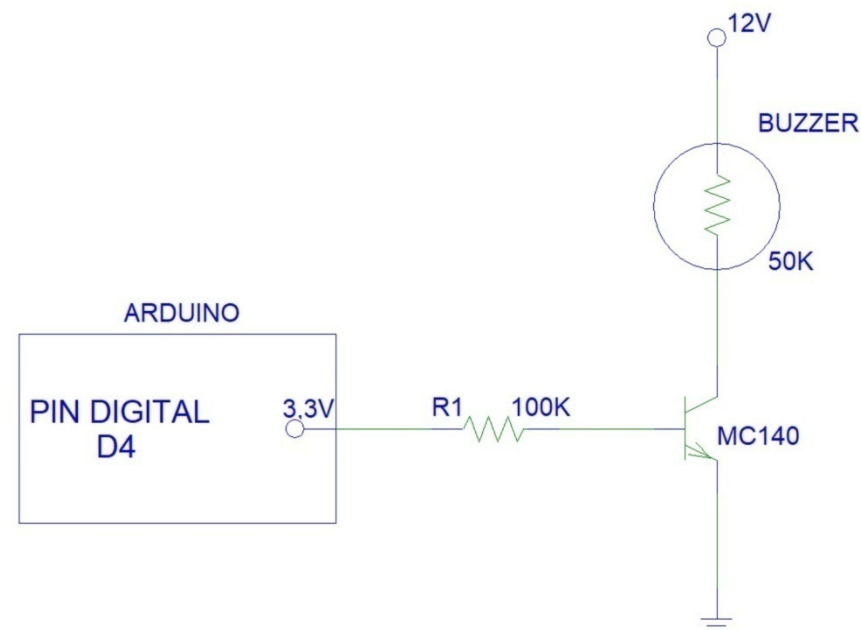


Figura 20: Circuit d'adaptació de l'Alarma sonora, amb el BUZZER.

$$I_b \cdot \beta > I_{c_sat} \quad Eq [11]$$

Perquè el transistor estigui en saturació $V_{ce}=0.2$ V. Per tant: $Eq [12] I_{c_sat} = \frac{V_{cc}-V_{ce}}{R_c} = \frac{12-0.2}{50 \cdot 10^3} = 2.36 \cdot 10^{-3} \rightarrow I_{c_sat} = \mathbf{0.236 \text{ mA}}$

Introduïm el valor de I_{c_sat} a l'equació i obtenim que : $I_b \cdot \beta > 0.236 \rightarrow I_b > \frac{0.236}{\beta} = \frac{0.236}{40} \rightarrow I_b > \mathbf{0.0059 \text{ mA}}$

A més, sin analitzem la malla: ($V_{ce}=0.7V$)

$$I_b = \frac{V_{bb} - V_{be}}{R_b} \quad Eq [12] \rightarrow R_b = \frac{V_{bb} - V_{ce}}{I_b} = \frac{3.3 - 0.7}{5.9 \cdot 10^{-6}} \rightarrow R_b = \mathbf{440.67 \text{ kOhms}}$$

El valor trobat de R_b és el valor màxim que pot tenir la resistència per tal que el transistor MC140 funcioni adequadament. Així doncs, perquè aquets no treballi al límit, establim una $R_b=100K$. Així, el valor de I_b serà: [Eq. 12]

$$I_b = \frac{V_{bb} - V_{be}}{R_b} = \frac{5 - 0.7}{100 \cdot 10^3} \rightarrow I_b = \mathbf{4.3 \cdot 10^{-5} = 0.043 \text{ mA}}$$

Comprovem que es compleix l'equació [11].

$$I_b \cdot \beta > I_{c_sat} \rightarrow 0.043 \cdot 40 > 0.236 \rightarrow \mathbf{1.72 \text{ mA}} > 0.236 \text{ mA}$$

Si posem una R_b de 100k el transistor MC140 funcionarà adequadament.

El pin utilitzat per realitzar aquesta alarma és el pin 4, el qual és un pin digital. El pin és d'aquest tipus, ja que sols hem d'enviar l'ordre de *high* o *low*, no un valor concret.

4.4 PLACA

Per poder fer possible la implementació del sistema el muntatge de tots els circuits mencionats es farà a una placa impresa, PCB. En aquesta, s'ha intentat optimitzar el màxim l'espai, fent que la placa quedés amb unes dimensions de 160x100 mm i dues cares.

A banda de tots els circuits mencionats, també s'han incorporat a la PCB dos interruptors, els quals ens donaran l'opció que l'usuari pugui configurar el sistema. El primer d'ells té la funcionalitat comú d'ON-OFF, per tal que el sistema pugui ser apagat si es considera oportú. Per altra banda, l'altre interruptor és el que ens permetrà estar en mode *run* o en mode *configuració*.

A més, per tal de fer-ho més visual, els dos interruptors van acompanyats de leds, de tal forma que si està en ON i mode *run*, s'il·luminen dos leds verds indicant que el sistema funciona correctament i que, per tant, es pot procedir a utilitzar-lo. Per contra, si s'encén un led verd i un vermell indica que està encès però en mode configuració, i que no està en disposició de ser utilitzat.

A més, per tal d'assegurar una connexió forta i duradora tant dels sensors com de la tensió d'entrada (V_{cc}), la connexió d'aquests amb la PCS es realitzarà mitjançant regletes atornillades.

A continuació a les figures 20 i 21, es pot veure el circuit general del sistema i la col·locació de cada element electrònic a la placa PCB.

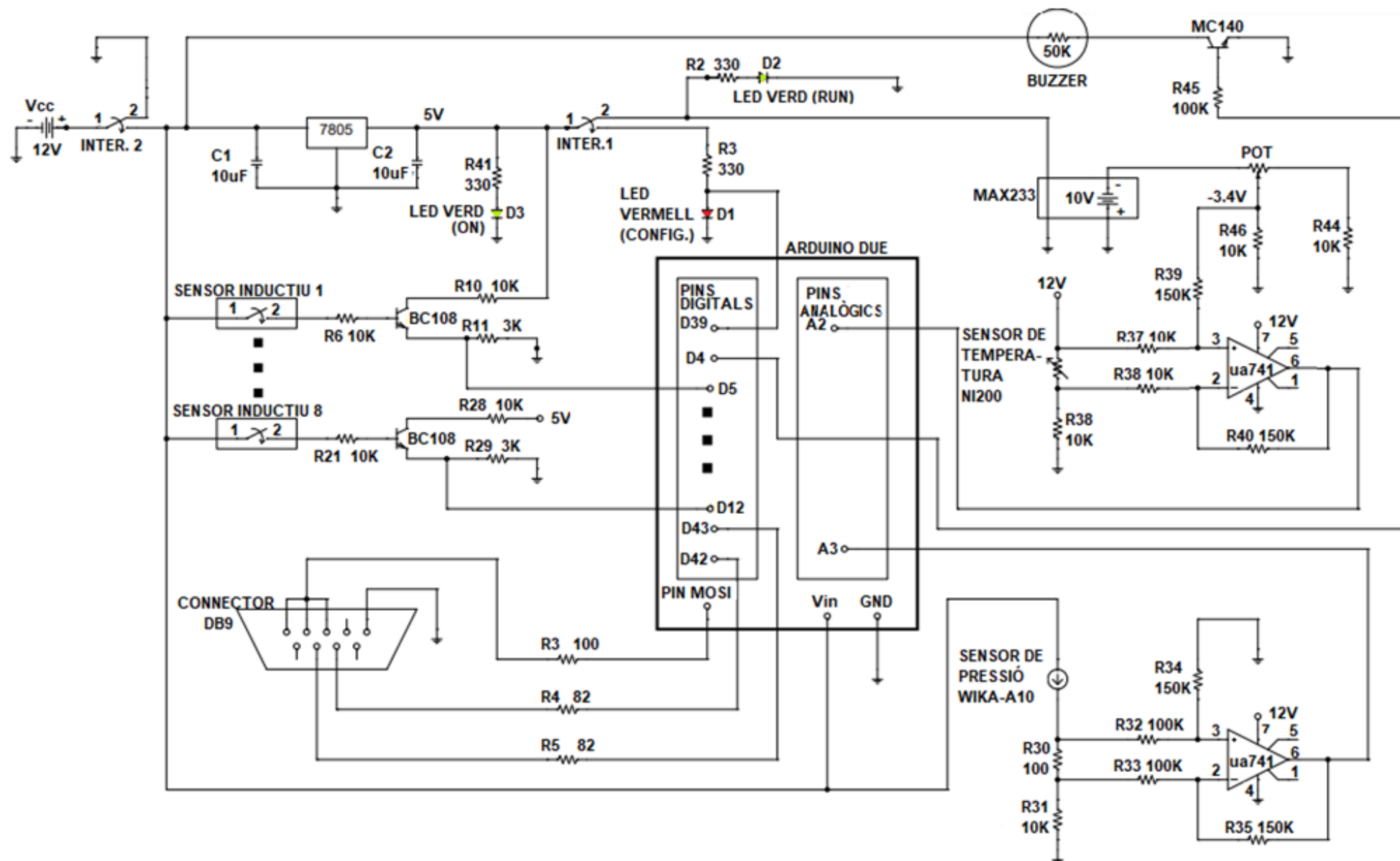


Figura 21: Circuit general del sistema

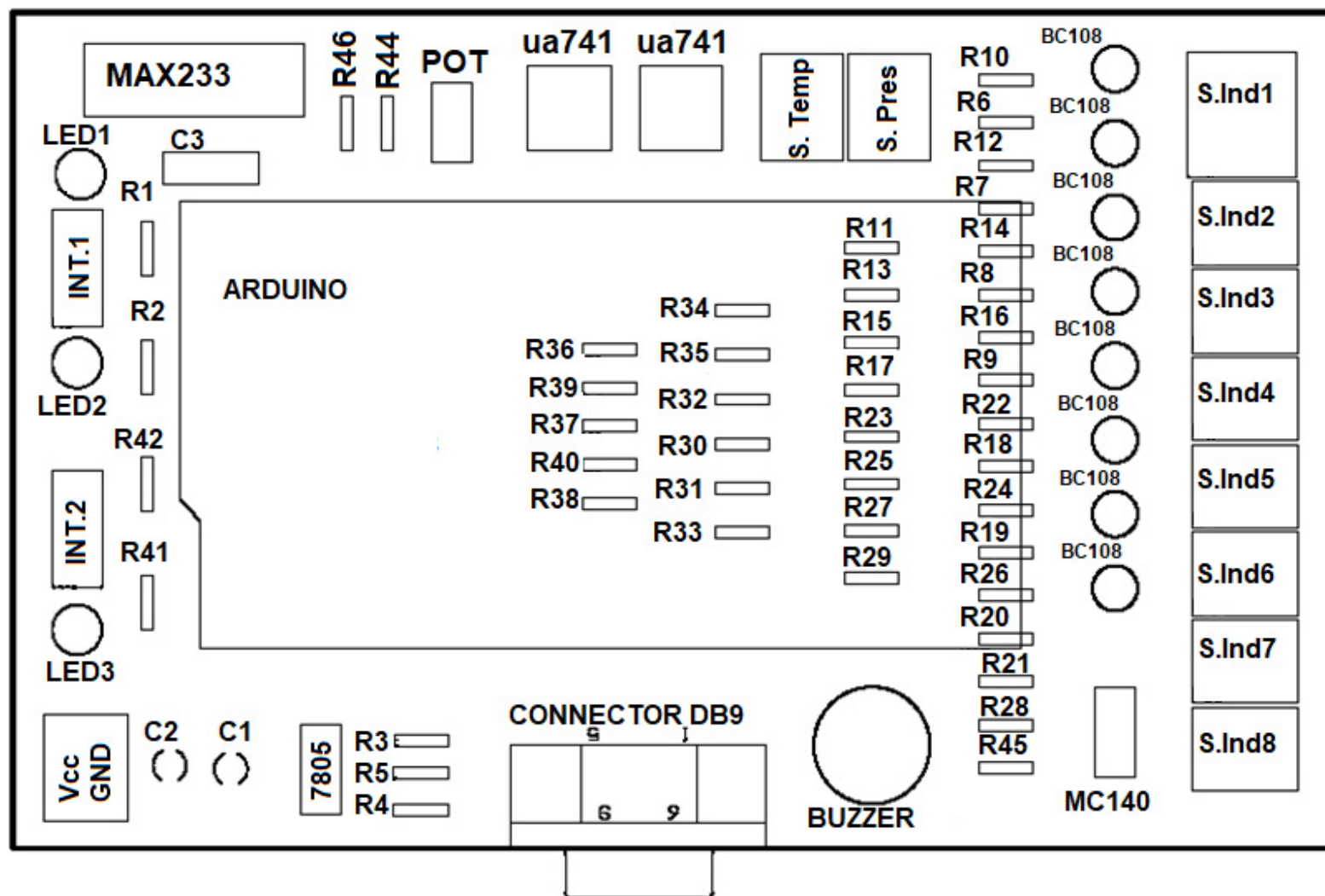


Figura 22: Ubicació dels elements a la placa PCB

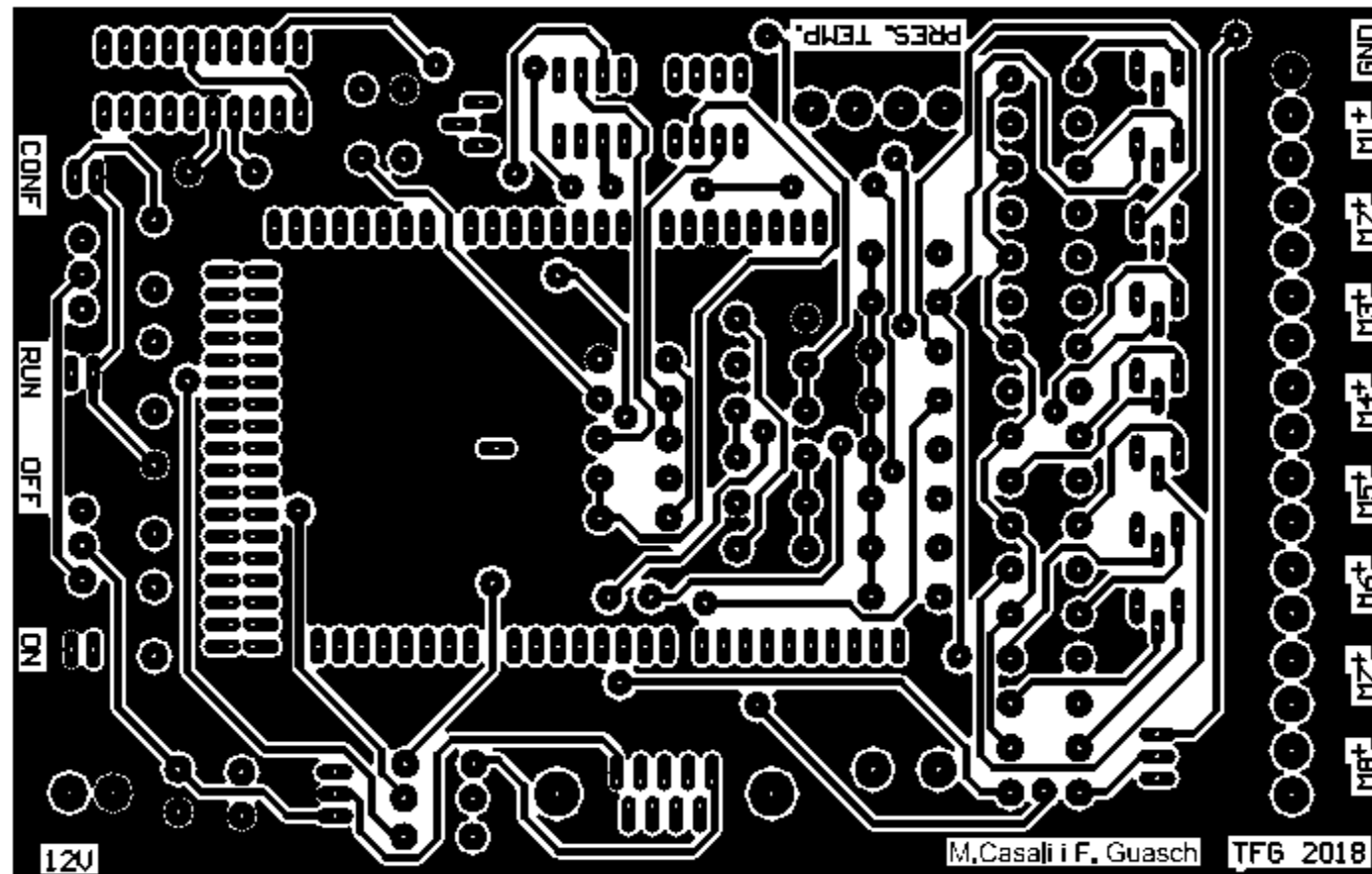


Figura 23: Layout placa cara superior

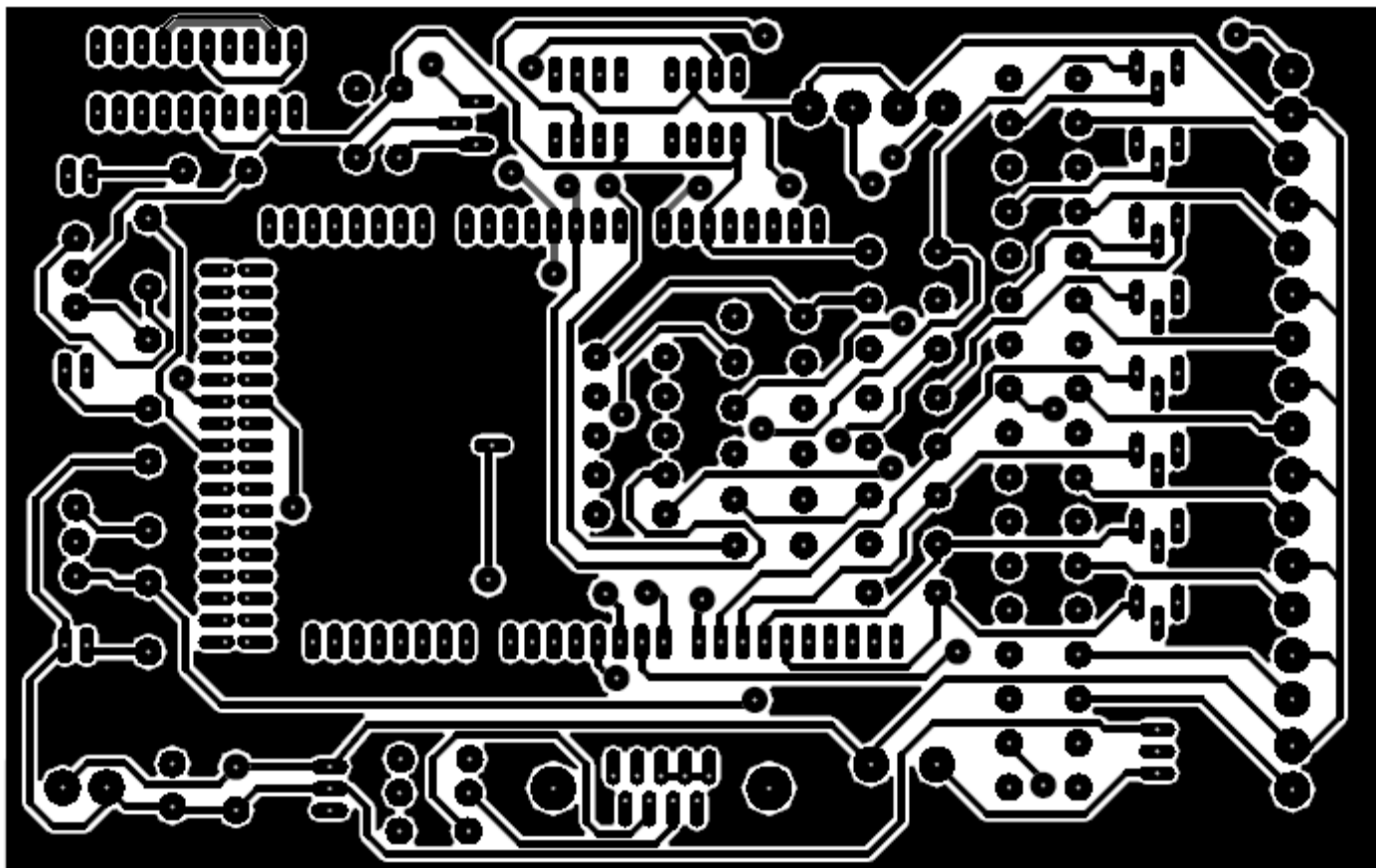


Figura 24: Layout placa cara inferior

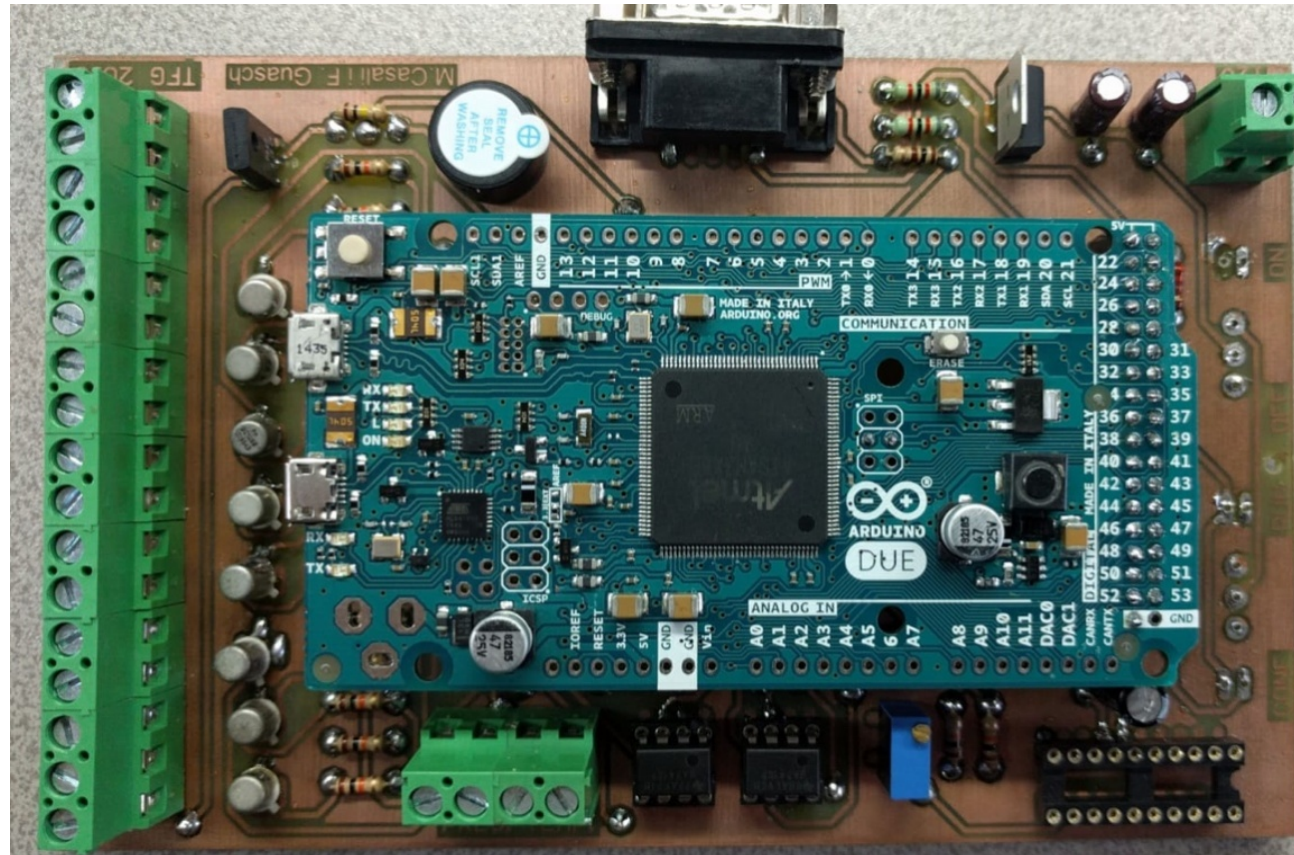


Figura 25: Placa PCB

5. SOFTWARE

5.1 PRINCIPAL FUNCÍO DEL PROGRAMA

Com hem mencionat en apartats anteriors, el sistema té un mode de *run*, on el sistema funcionarà normalment, i un mode de configuració, en el qual l'usuari podrà canviar els paràmetres dels límits de velocitat, pressió i temperatura.

Quan es posi el botó d'ON-OFF en actiu, el programa s'iniciarà. Si el segon interruptor està en mode run, el programa executarà el subprograma que anomenarem principal. Si pel contrari, el segon interruptor està en mode configuració, el programa executarà un subprograma, el qual anomenarem programa de configuració. El programa es basa en un sistema d'interrupcions i consta de dues parts principals, que anomenarem mode *configuració* i mode *run*.

La primera vegada que l'usuari faci ús del sistema de sensat, ha de posar el mode configuració ja que els valors per defecte de límits tenen un valor de 0.

El mode de configuració, requereix de l'ús d'un ordinador per tal de que l'usuari introdueixi els valors desitjats com a límits de revolucions per minut, temperatura i pressió. Per això, la placa es comunica mitjançant línia sèrie amb el programa PuTTY de l'ordinador. Aquest programa és l'encarregat de demanar a l'usuari els valors que desitja i guardar-los.

Per evitar la pèrdua d'aquestes dades una vegada desconnectem l'Arduino del PC, aquestes són emmagatzemades a la memòria flash del microcontrolador.

El mode *run*, o mode de treball, és l'encarregat de llegir i processar les dades proporcionades per els sensors. Aquest, està dividit en 5 subfuncions i 9 interrupcions.

Hi ha 8 interrupcions (una per cada sensor inductiu), aquestes són les encarregades de comptar, les vegades que els sensors detecten.

L'altre interrupció requereix dels valors emmagatzemats als comptadors per les interrupcions anteriors. Aquesta s'activa una vegada cada dos segons (a petició de l'usuari), per a guardar el valor dels comptadors i resetejar-los.

Previ a les subfuncions del mode *run*, hi ha una condició, per llegir els valors guardats a la memòria Flash al mode de configuració i guardar-los en una nova variable. Mitjançant aquesta condició tan sols es llegiran de la memòria Flash la primera vegada que s'activi el mode *run*.

La subfunció de les revolucions agafa les variables guardades de l'anterior interrupció i les multiplica per 30 per obtenir les revolucions per minut.

La subfunció de temperatura i la de pressió segueixen el mateix esquema. Llegeixen l'entrada analògica corresponent de l'Arduino. La divideixen entre 1023 i la multipliquen per el valor màxim de tensió de la placa (3.3V) per obtenir el voltatge corresponent als bits llegits. Aquest valor de tensió, mitjançant la recta de temperatura envers voltatge i la de pressió envers voltatge, ens dona el valor actual tan de pressió com de temperatura.

L'última subfunció del mode *run* és la que fa funcionar les alarmes del sistema. Primerament connecta amb el monitor on es veuran les alarmes i mostra constantment el valor actual de les rpm, pressió i temperatura i el límit introduït en el mode de configuració, Mitjançant un conjunt de condicions, si el valor actual supera el valor límit, apareix un missatge d'alarma indicant la part que no funciona correctament. Conjuntament a la alarma visual hi ha una alarma sonora generada a través de l'Arduino, mitjançant una sèrie de polsos enviats al Buzzer.

Per tal d'entendre correctament el funcionament general d'aquest, a continuació podrem trobar el seu pseudocodi.

```

1 PSEUDOCODI PROGRAMA GENERAL
2
3 if ( SELECTOR MODE == 1)
4 {
5     Mode Configuracion ()
6     {
7         Introduir velocitat motor1
8         Introduir velocitat motor2
9         Introduir velocitat motor3
10        Introduir velocitat motor4
11        Introduir velocitat motor5
12        Introduir velocitat motor6
13        Introduir velocitat motor7
14        Introduir velocitat motor8
15        Introduir velocitat PressioMaxima
16        Introduir velocitat PressioMinima
17        Introduir velocitat Temperatura Maxima
18    }
19 }
20 else
21 {
22     Interrupcio contadors ()
23     {
24         INTMOTOR1, RISING
25         valor Contador1 ++
26         INTMOTOR2, RISING
27         valor Contador2 ++
28         INTMOTOR3, RISING
29         valor Contador3 ++
30         INTMOTOR4, RISING
31         valor Contador4 ++
32         INTMOTOR5, RISING
33         valor Contador5 ++
34         INTMOTOR6, RISING
35         valor Contador6 ++
36         INTMOTOR7, RISING
37         valor Contador7 ++

```

```

38         INTMOTOR8, RISING
39         valor Contador8 ++
40     }
41     Interrupcio timer()
42     Timer = 2segons
43     MC1 = valor Contador1
44     MC2 = valor Contador2
45     MC3 = valor Contador3
46     MC4 = valor Contador4
47     MC5 = valor Contador5
48     MC6 = valor Contador6
49     MC7 = valor Contador7
50     MC8 = valor Contador8
51     Reset valor contador
52     Reset Timer
53
54     Mode Programa ();
55     {
56         Calcul rpm ( M1 = MC1 * 60)
57         Calcul rpm ( M2 = MC2 * 60)
58         Calcul rpm ( M3 = MC3 * 60)
59         Calcul rpm ( M4 = MC4 * 60)
60         Calcul rpm ( M5 = MC5 * 60)
61         Calcul rpm ( M6 = MC6 * 60)
62         Calcul rpm ( M7 = MC7 * 60)
63         Calcul rpm ( M8 = MC8 * 60)
64
65         Vt = lectura pin Analogic
66         Calcul Temperatura ( temperatura = Y + M * Vt)
67         Vp = lectura pin Analogic
68         Calcul Pressio ( pressio= Y + M * Vp)
69
70         if ( M1 > motor1 )
71         {
72             Alarma visual
73             Alarma sonora

```

Quan el programa executi el subprograma de configuració, l'usuari haurà d'anar introduint els valors que parametriza el programa principal. Així doncs, un cop es tria l'opció configuració, per pantalla es mostrarà un missatge de "Valor motor 1: ". Un cop mostrat aquest missatge, l'usuari haurà d'introduir el valor que desitgi. Un cop introduït el valor, el programa passarà a demanar el següent paràmetre, fins a completar la velocitat dels 8 motors, la pressió màxima i mínima, i la temperatura màxima.

Per altra banda, si el segon interruptor està en mode *run*, s'executarà el programa principal. Aquest serà l'encarregat de gestionar les dades i fer sonar les alarmes. És a dir, serà l'encarregat que el sistema funcioni com s'espera.

Aquest programa principal, mentre s'estigui executant, anirà avaluant les lectures de les dades amb la dinàmica que mostra el diagrama de flux de la figura 23.

Aquestes dades, siguin de revolucions, temperatura o pressió, seran comparades amb els valors que s'han introduït en el subprograma de configuració. Si aquestes dades adquirides estan dins els paràmetres establerts, el programa tornarà a fer lectura de dades. Per contra, si

un d'aquests paràmetres no és l'adequat, el programa, automàticament, farà sonar l'alarma sonora i, a més, mostrarà per pantalla quin és el paràmetre que està fora de rang.

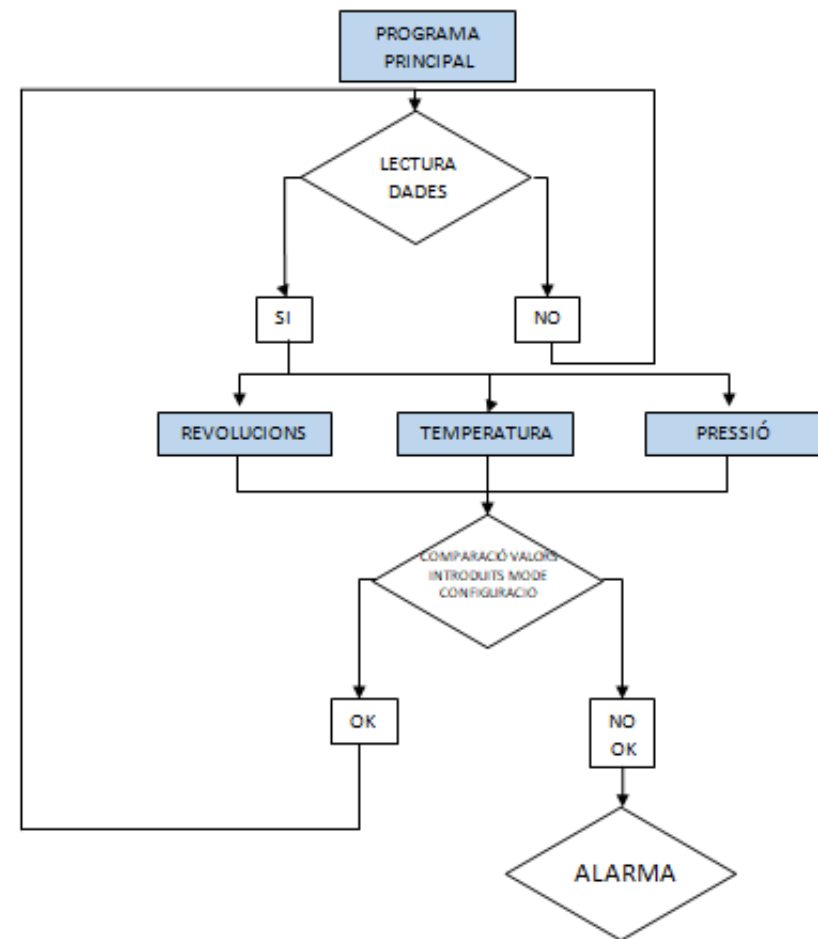


Figura 26: Diagrama de flux del programa principal

5.2 LLIBRERIES

5.2.1 ARDUINO TVOout

Aquesta llibreria vam decidir utilitzar-la primerament per connectar el cable RCA a l'Arduino i generar un senyal de vídeo.

Va quedar descartada en trobar-nos amb incompatibilitats a la placa ArduinoDue, ja que hi ha 2 pins de funció digitals (#7 i #9) que la seva funció és diferent en la Uno i en la Due. Vam intentar modificar la llibreria per no haver d'utilitzar els pins #7 i #9, però implicava unes modificacions de codi amb una gran dificultat. Vam decidir, doncs, descartar aquest mètode de connexió a la pantalla i aquesta llibreria.

5.2.2 VGA DUE

Aquesta llibreria és específica per a la placa ArduinoDue, permet generar un senyal de vídeo composta VGA o NTSC/PAL des de l'Arduino. Hi ha 3 tipus de senyal que es poden generar:

- Senyal VGA monocromàtica.
- Senyal VGA a color.
- Senyal NTSC/PAL.

En el nostre cas vam descartar el senyal NTSC/PAL perquè no volíem transmetre un senyal de vídeo analògic de televisió. Vam escollir la monocromàtica, ja que no era necessari el canvi de color.

Un cop definit el tipus de senyal ens vam centrar en la resolució, la màxima que suporta en monocromàtic és de 800x600.

VGA.begin(x, y);

(S'ha de posar al voidsetup, per a inicialitzar la llibreria els valors de x i y són la resolució de la pantalla)

intVGA.begin(int x, int y, int m=VGA_MONO);

(Aquesta funció permet introduir la resolució desitjada a la pantalla, següent X la variable horitzontal i Y la variable vertical, la variable m, fa referència al tipus d'imatge que generarem, en el nostre cas VGA_MONO, ja que serà monocromàtica).

VGA.clear(int col = 0);

(Aquesta funció, neteja la pantalla al color definit per el paràmetre col, en aquest cas si no definim cap paràmetre en color negre).

VGA.setPrintWindow(int x0,int y0,int x1,int y1);

(Aquesta funció limita l'àrea de treball de la pantalla, en l'àrea definida per els paràmetres x0, x1 i y0 i y1).

VGA.fillRect(int x0,int y0,int x1,int y1,int col);

VGA.drawRect(int x0,int y0,int x1,int y1,int col);

(Aquestes dues funcions serveixen per generar el contorn d'un rectangle, la segona, i la primera omple el rectangle, els paràmetres x0 i y0 indiquen la coordenada d'inici del rectangle i els paràmetres x1, y1, i el paràmetre col, representa el color del rectangle, en el nostre cas un 1 que es el color blanc).

VGA.print(X);

VGA.println(X);

(Aquestes dues funcions funcionen igual que les dues funcions de la línia sèrie, Serial.print(X) i Serial.println(X)).

5.2.3 DueTimer

Aquesta llibreria és necessària per controlar els timers de l'ArduinoDue, ja que les funcions generals de control dels timers, no venen controlades amb les llibreries de sèrie de l'Arduino Due.

Un cop instal·lada hem utilitzat aquesta llibreria per a programar una interrupció per temps, perquè cada segon salti una interrupció en la qual es guarda el valor del comptador de revolucions (el nombre de polsos que han tingut els sensors capacitius durant aquest temps).

D'aquesta llibreria hem utilitzat les següents funcions:

```
Timer3.attachInterrupt(TIMERINT).setFrequency(1).start();
```

Aquesta funció ens crida l' interrupció INTMOTORS, 1 vegada cada segon (amb el valor introduït a (setFrequency(X)).

L'hem utilitzat perquè activi la interrupció INTMOTORS, la qual ens guarda el valor dels comptadors associats als sensors capacitius per contar revolucions, els guarda en una altra variable i els posa a 0 de nou, perquè aquests s'actualitzin una cop per segon.

5.3 FUNCIONS

5.3.1 FUNCIO DE CONFIGURACIO

En aquesta gran funció del programa Arduino, l'usuari introdueix els valors límit que desitja per a la velocitat de gir dels motors, pressió i temperatura.

Per llegir aquestes dades es fa a través de l'ordinador, i el programa PUTTY (client SSH, Talnet, rlong, TCP raw amb llicència lliure). Mitjançant aquest terminal, el programa preguntarà a l'usuari les diferents dades que necessita.

Dins d'aquesta funció ens trobem amb els textos que demanen la introducció de dades i dues petites subfuncions:

1. Lectura ()

Primerament hi ha un bucle d'espera per què arribi un valor per la línia sèrie, en cas que hi hagi un valor, el llegeix com a una cadena de nombres (str) que després convertim a un nombre *float* (data). Guardem aquest nombre en una variable (*valorIntroduit*). I per finalitzar debuguem el valor.

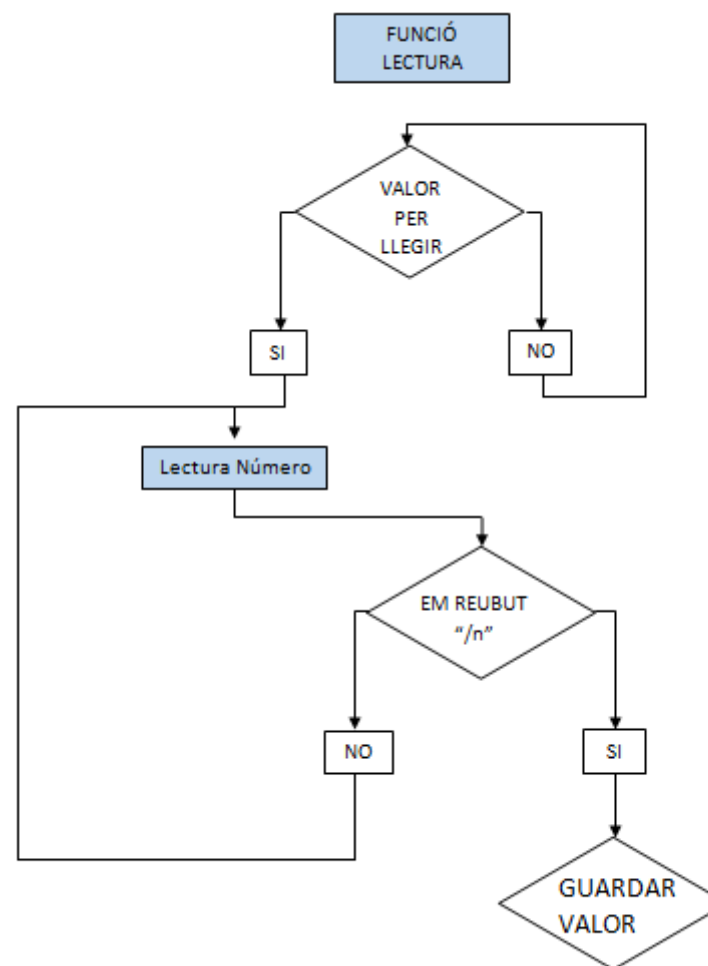


Figura 27: Diagrama de flux de la funció lectura.

```
void lectura () {  
  
    while (Serial.available() == 0)  
    {  
        //bucle de espera a llegir un valor per linia serie  
    }  
    if (Serial.available() > 0)  
    {  
        String str = Serial.readStringUntil('\n');  
        float data = str.toFloat();  
        float c = data;  
        valorIntroduit = c;  
        Serial.println(c);  
        DEBUG(data);  
    }  
}
```

Figura 28: Implementació de la funció "Lectura".

2. resetValor ()

En aquesta funció resetegem la variable guardada (valorIntroduit) després de llegir la línia sèrie.

```
void resetvalor () {      //funcio per resetejar el valor qe llegim per linia serie.  
    valorIntroduit = 0;  
}
```

Figura 29: Implementació de la funció "resetValor"

5.3.2 FUNCIO RUN

Aquesta és la gran funció en la qual es basa el funcionament de tot el sistema de sensors dissenyat. Es basa en la funció i el conjunt d'interrupcions que llegeixen els valors dels sensors, i mostren els missatges d'alarma.

Primerament analitzarem les interrupcions:

1. INTMOTOR X ()

Aquest conjunt de 8 interrupcions corresponen als sensors inductius adjunts als 8 motors de la collitadora.

Aquestes interrupcions simplement augmenten el valor d'una variable (l'anomenarem comptador) en 1, cada vegada que els sensors detecten que el motor dona 1 volta (una revolució).

```
void INTMOTOR1 () {  
  s ++;  
}  
void INTMOTOR2 () {  
  t ++;  
}  
void INTMOTOR3 () {  
  u ++;  
}  
void INTMOTOR4 () {  
  v ++;  
}  
void INTMOTOR5 () {  
  w ++;  
}  
void INTMOTOR6 () {  
  x ++;  
}  
void INTMOTOR7 () {  
  y ++;  
}  
void INTMOTOR8 () {  
  z ++;  
}
```

Figura 30: Implementació de la funció INTMOTOR X ()

2.TIMERINT ()

Aquesta interrupció s'activa mitjançant el timer 3, 1 vegada per segon. En aquesta interrupció es guarden els valors dels comptadors (interrupció anterior), sempre que el valor del T sigui igual a 1, on $T = T + 1$. Aquesta condició ens assegura que sols es faci la conta 1 vegada per segon.

Un cop guardats els valors dels sensors capacitius en les noves variables (C1, C2...), es posen a 0 els comptadors i la variable T. En el cas que la suma de $T = T + 1$, resultés en un nombre més gran de 1 no es fa res durant la interrupció.

Entrem, ara, a la funció principal del programa: *run*.

RUN ()

Aquesta funció agrupa totes les diferents funcions que permeten la comparació de les variables introduïdes per l'usuari i les llegides pels diferents sensors.

Revolucions ()

En aquesta subfunció, es multipliquen els valors de les variables (C1, C2....) corresponents al comptador del sensor capacitius, per 60, ja que hem comptat quantes vegades s'activa o no el sensor en 1 segon per tal de treure el número de revolucions per minut.

```
void TIMERINT () { // aquesta interrupció entrara 1 vegada cada
    T = T + 1;
    if (T == 1) { //ens asegurem de que sols entra 1 vegada ca
        C1=s; //guardem el valor del contador 1
        C2=t; //guardem el valor del contador 2
        C3=u; //guardem el valor del contador 3
        C4=v; //guardem el valor del contador 4
        C5=w; //guardem el valor del contador 5
        C6=x; //guardem el valor del contador 6
        C7=y; //guardem el valor del contador 7
        C8=z; //guardem el valor del contador 8

        T = 0; //Resetegem el temps
        s=0; //Resetegem el contador1
        t=0; //Resetegem el contador2
        u=0; //Resetegem el contador3
        v=0; //Resetegem el contador4
        w=0; //Resetegem el contador5
        x=0; //Resetegem el contador6
        y=0; //Resetegem el contador7
        z=0; //Resetegem el contador8
    }
    else
    {
        //buit
    }
}
```

Figura 31: Implementació de la funció "TIMERINT"

```
void revoluciones () {  
  M1 = C1 * 60;  
  M2 = C2 * 60;  
  M3 = C3 * 60;  
  M4 = C4 * 60;  
  M5 = C5 * 60;  
  M6 = C6 * 60;  
  M7 = C7 * 60;  
  M8 = C8 * 60;  
}
```

Figura 32: Implementació de la funció "Revolucions"

Temperatura ()

En aquesta subfunció, calculem la temperatura, segons el valor de tensió que ens arriba a l'Arduino a través del pin analògic A2. Per fer-ho multipliquem el valor analògic llegit de l'esmentat pin, el dividim entre el valor màxim de bits que es poden llegir (1023) i el multipliquem pel valor màxim de tensió que podem tenir al pin 3.3V.

Una vegada obtenim aquest valor de tensió mitjançant la recta de temperatura envers voltatge, multiplicant el valor obtingut pel pendent de la recta (M_t) i sumant-li el desplaçament vertical que pugui haver terme (XX).

```
void Temperatura ()  
{  
  
    Vt = (analogRead(SensorTemperatura)/1023.0)*3.3; // Volatge que arriba al pin analogic A2, ho pasem a bits  
    Temp = XX + Mt * Vt;  
}
```

Figura 33: Implementació de la funció "Temperatura"

Pressió ()

En aquesta subfunció, calculem la pressió, segons el valor de tensió que ens arriba a l'Arduino a través del pin analògic A3. Per fer-ho multipliquem el valor analògic llegit de l'esmentat pin, el dividim entre el valor màxim de bits que es poden llegir (1023) i el multipliquem pel valor màxim de tensió que podem tenir al pin 3.3V.

Una vegada tenim aquest valor de tensió mitjançant la recta de pressió envers voltatge, multiplicant el valor obtingut pel pendent de la recta (Mp) i sumant-li el desplaçament vertical que pugui haver terme (XY).

```
void Pressio ()  
{  
    Vp = (analogRead(SensorTemperatura)/1023.0)*3.3; // Volatge que arriba al pin analogic A3, ho pasem a bits  
    Presion = XY + Mp * Vp;  
}
```

Figura 34: Implementació de la funció "Pressió"

5.3.3 FUNCIONS PER GUARDAR VARIABLES NO VOLÀTILS A LA MEMÒRIA FLASH

Un cop introduït el valor que desitgem de velocitat del motor i l'Arduino l'ha rebut, com aquest valor és més gran d'un byte (255 bits), (ja que sol oscil·lar entre les 1500 i les 2000 revolucions) hem dividit la variable introduïda entre 100 per obtenir un nombre entre el 0 i el 100, fen possible que es guardi com a un byte a la memòria flash.

La principal funció que hem utilitzat és la de guardar dades a la memòria:

```
Serial.println( "INTRODUIR VELOCITAT DE GIR (rpm) DEL MOTOR 6");  
lectura ();  
motor6=valorIntroduit;  
Y = motor6 /100;  
dueFlashStorage.write(X,Y);
```

Figura 35: Implementació de la funció per guardar dades a la memòria flash.

On X equival a la direcció de memòria utilitzada i Y al byte que desitgem guardar.

En el cas de guardar el valor de temperatura o de les pressions mínima i màxima, no es necessari dividir entre 100 per tal d'obtenir un valor que es pugui emmagatzemar en un byte, ja que la temperatura utilitza un rang de 0 a 100 °C i la pressió un rang de 0 a 10 bars.

Per llegir les dades emmagatzemades a la memòria, hem realitzat una condició perquè no entri a llegir de la memòria, si ja ha entrat prèviament.

5.3.4 FUNCIONS DE LES ALARMES

Primerament netegem qualsevol imatge residual que pugui aparèixer a la pantalla, generem un rectangle blanc en el que apareixeran les alarmes, i escrivim els valors de les variables dels motors, temperatura i pressió a temps real.


```
void Pantalla () {           //funció per ,pstrar els valors per pantalla
    VGA.clear();             //neteja la imatge de la pantalla
    int x0=190,y0=0,x1=320,y1=240;    //coordenades del rectangle blanc on apareixeran les alarmes
    VGA.setPrintWindow(0,0,320,240);  //funcio que estableix els limits de la zona de treball de la pantalla
    VGA.fillRect(x0,y0,x1,y1,1);      //rectangle blanc de les alarmes
    VGA.drawRect(x0,y0,x1,y1,1);

    VGA.print("revoluciones motor1:");
    VGA.println(M1);
    VGA.print("revoluciones motor2:");
    VGA.println(M2);
    VGA.print("revoluciones motor3:");
    VGA.println(M3);
    VGA.print("revoluciones motor4:");
    VGA.println(M4);
    VGA.print("revoluciones motor5:");
    VGA.println(M5);
    VGA.print("revoluciones motor6:");
    VGA.println(M6);
    VGA.print("revoluciones motor7:");
    VGA.println(M7);
    VGA.print("revoluciones motor8:");
    VGA.println(M8);
    VGA.print("Pressio");
    VGA.println(Presion);
    VGA.print("Temperatura");
    VGA.println(Temp);
}
```

Figura 36: Implementació de la funció per mostrar els valors per pantalla.

Amb aquesta funció introduïm l'alarma sonora, aquesta funciona mitjançant, una funció *if* que s'activarà si el temps transcorregut és major o igual a l'interval definit en el que volem que soni l'alarma. Si entra el bucle, modificarà l'estat del pin del buzzer, i així successivament mentre estigui la funció activa.

```
void alarma_sonora ()
{
  unsigned long currentMillis = millis();
  if(currentMillis - previousMillis >= interval){      //guardem el ultim interval del alarma sonora
    previousMillis = currentMillis;

    if(BuzzerState == LOW) {      //si el buzzer esta encs el posa en marxa o vicesersa
      BuzzerState = HIGH;
    }
    else {
      BuzzerState = LOW;
    }
    digitalWrite(Buzzer, BuzzerState);    //escriu al pin el valor de la variable de estat del buzzer
  }
}
```

Figura 37: Implementació de la funció de l'alarma sonora.

```
if (motor8 == 0)
{
    PressioMaxima = dueFlashStorage.read(16);
    PressioMinima = dueFlashStorage.read(18);
    Temperatura = dueFlashStorage.read(20);
    motor1 = dueFlashStorage.read(0)*100;
    motor2 = dueFlashStorage.read(2)*100;
    motor3 = dueFlashStorage.read(4)*100;
    motor4 = dueFlashStorage.read(6)*100;
    motor5 = dueFlashStorage.read(8)*100;
    motor6 = dueFlashStorage.read(10)*100;
    motor7 = dueFlashStorage.read(12)*100;
    motor8 = dueFlashStorage.read(14)*100;
}
```

Figura 38: Implementació de la funció de lectura de la memòria flash.

En aquest cas, per a la lectura tan sols hem de posar l'adreça de memòria que volem llegir. En el cas dels motors, com anteriorment em dividit entre 100 per tal que es poguessin emmagatzemar en un byte, ara toca tornar al valor original.

5.4 CONNEXIÓ DEL TERMINAL

Teníem tres opcions perquè l'usuari és connecti a l'Arduino per introduir els paràmetres de configuració. Mitjançant el programa Arduino, el Hyperterminal de Windows o el programa PuTTY.

La primera d'aquestes opcions, la més senzilla, va quedar descartada perquè si l'usuari no coneix el sistema Arduino, podia esborrar línies de programa, i/o modificar paràmetres. Per evitar aquestes complicacions vam descartar aquest sistema.

El hyperterminal i el PuTTY són dos programes de terminal de TCP (Transmission Control Protocol), aquests permetran rebre i enviar dades per la línia sèrie de l'Arduino, especificant el port de l'ordenador i la velocitat.

Finalment degut a que el sistema PuTTY està internacionalitzat i que el Hyperterminal de Windows ha estat retirat i es considera obsolet per part de Microsoft, vam decidir continuar amb el sistema PuTTY.

5.4.1 CONFIGURACIÓ PUTTY

Com ja em dit el PuTTY és un client de terminal TCP (Transmission Control Protocol), de programari lliure disponible a la seva pròpia web ^[19].

Per configurar-lo requereix la introducció del port COM on està situada la placa Arduino; per veure-ho s'ha d'anar Mi PC, botó dret del ratolí i Administrar (imatge 36).

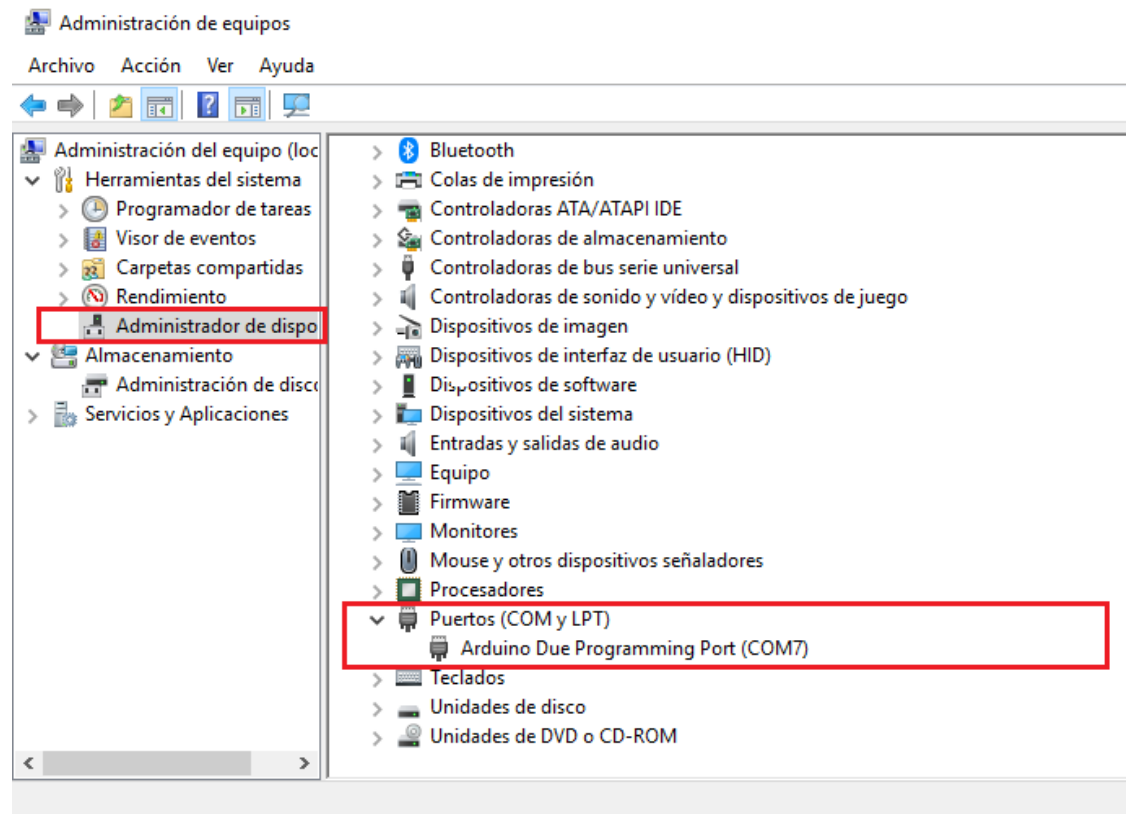


Figura 39: Connexió del port COM de l'Arduino.

S'obrirà la següent pantalla, figura 36, en la qual ens mostra a quin port està situat l'Arduino. Després s'ha d'especificar la velocitat de la línia sèrie. Per defecte nosaltres al programa Arduino l'hem inicialitzat a 9600 bauds.

Per tant introduïm les dades i podem guardar el perfil de l'Arduino per no haver d'introduir-ho tot cada vegada, com es mostra a la figura 37.

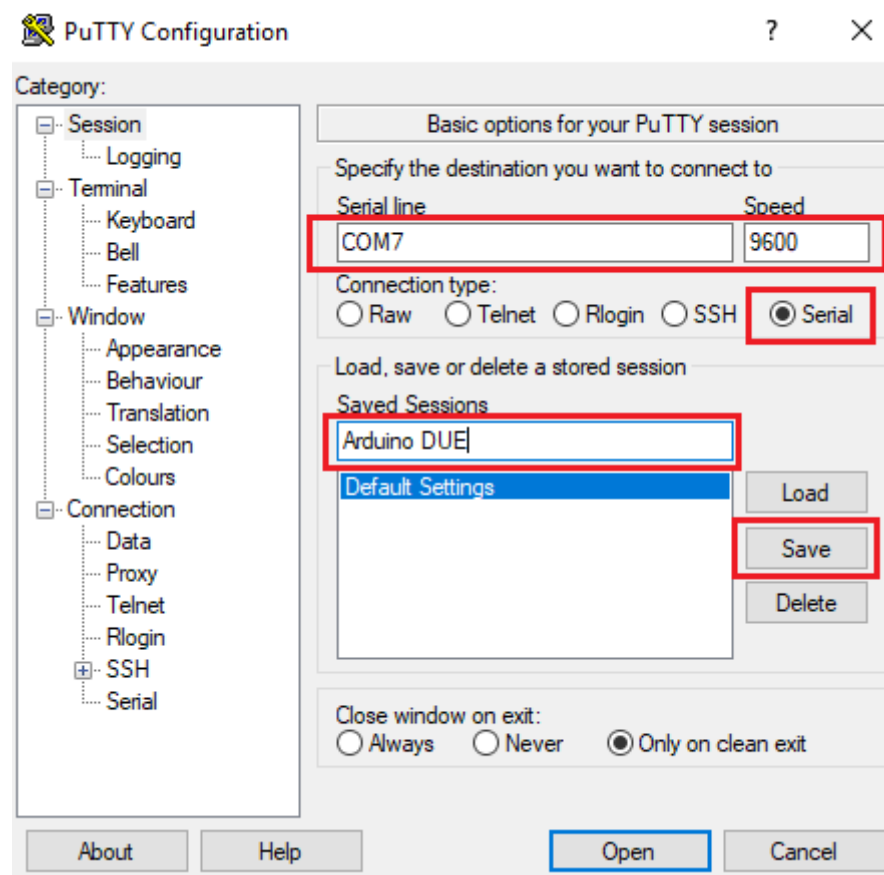


Figura 40: Guardar sa configuració de l'Arduino.

Amb tot això ja estaria connectat l'Arduino, amb el client PuTTY tan sols faltaria fer clic al botó de OPEN, i se'ns obrirà una nova finestra en negre a esperar rebre dades per la línia sèrie.

6. MANUAL DE FUNCIONAMENT DEL SISTEMA PER L'USUARI

En aquest punt, trobem un dossier, que fa de manual d'ús per al usuari de la collitadora. Està fet de forma que l'usuari el pugui tenir imprès per facilitar la seva consulta.

En ell s'explica com connectar els sensors a la placa, mitjançant fotografies. També té un apartat de com descarregar, instal·lar i configurar el client PuTTY, per connectar-lo amb la placa Arduino i configurar els paràmetres desitjats per al sistema. Finalment el manual consta d'un petit resum del funcionament de les alarmes, amb un exemple visual de quin aspecte tindran aquestes en el monitor.

Manual d'us del sistema de sensorat

Margarita Casalí & Ferran Guasch
12/07/2018

UPC (EPSEVG)

Índex

1.CONNEXIÓ DE LA PLACA I L'ARDUINO.....	72
2.CONFIGURACIÓ PUTTY.....	74
3.FUNCIONAMENT DE LES ALARMES.	79
Velocitat:	79
Temperatura:	79
Pressió:	79
4.PROGRAMA ARDUINO.....	81

1.Connexió de la placa i l'Arduino.

En aquest apartat indicarem com es connecten els diversos sensors a la placa PCB i com es connecta l'arduino en cas de treure-la per reprogramar-la.

El sensor de Pressió consta de dos cables, positiu i negatiu. El positiu és de color vermell, i el negatiu de color blau, i s'han de connectar segons la figura 38.

El sensor de Temperatura consta de dos cables, marró i blau; al funcionar com una resistència variable, és indiferent on connectem cadascun.

Els sensors Inductius, consten de 3 fils: un positiu, un negatiu i un de senyal. El negatiu de cada sensor s'ha de connectar mitjançant una regleta a un comú i, després, a l'últim connector, com es mostra a la figura 38. El fil de senyal i el positiu s'han de connectar segons la mateixa figura.

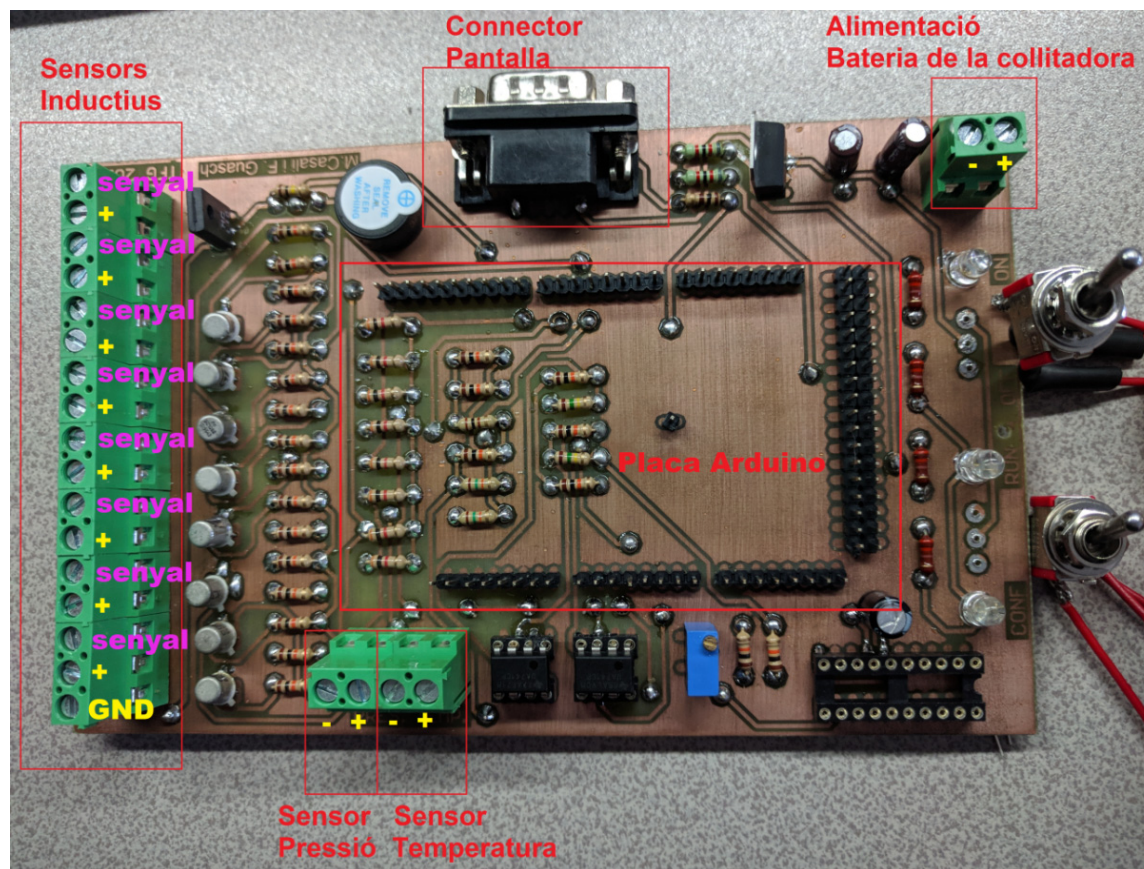


Figura 41: Connexió dels sensors a la placa PCB.

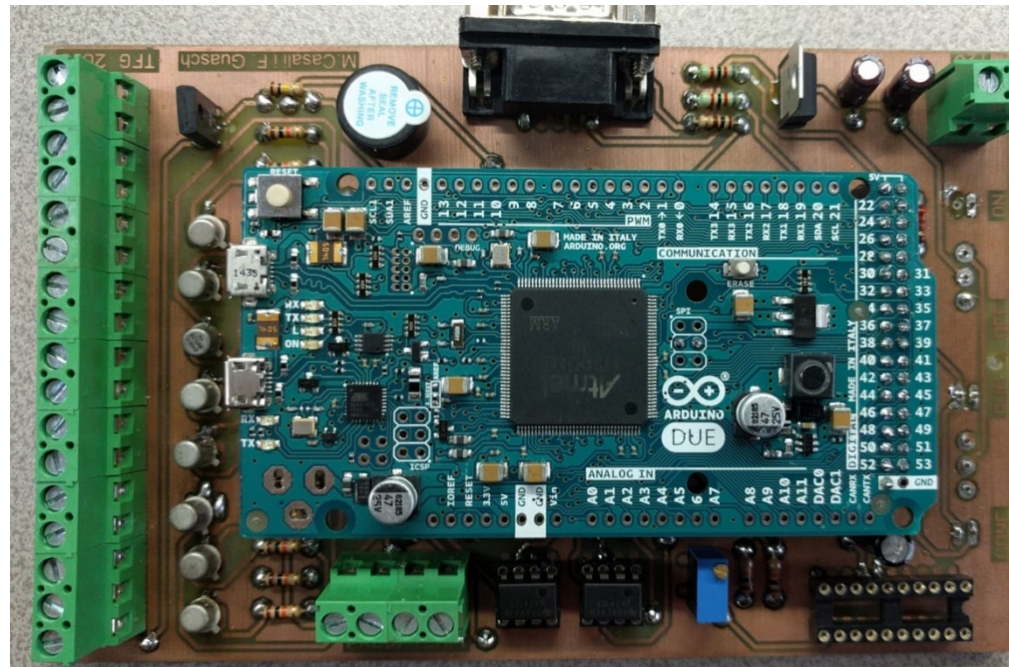


Figura 42: Placa PCB amb el microcontrolador Arduino

Per a la col·locació de la placa Arduino, senzillament s'han de fer coincidir els pins amb els connectors femella situats a la placa, el pin central del MOSI, comporta una petita dificultat de connexió. Un cop situada la placa Arduino ens quedaria com es mostra a la figura 39.

2. Configuració PUTTY

Com configurar el terminal PUTTY per introduir les dades de configuració a la Arduino.

1r) Per descarregar el programa PUTTY (En cas de ja tenir instal·lat el PUTTY, procedeixi al pas 2).

Anar a la web : <https://putty.org/> i descarregar el programa, fent clic on s'indica a la fotografia següent:

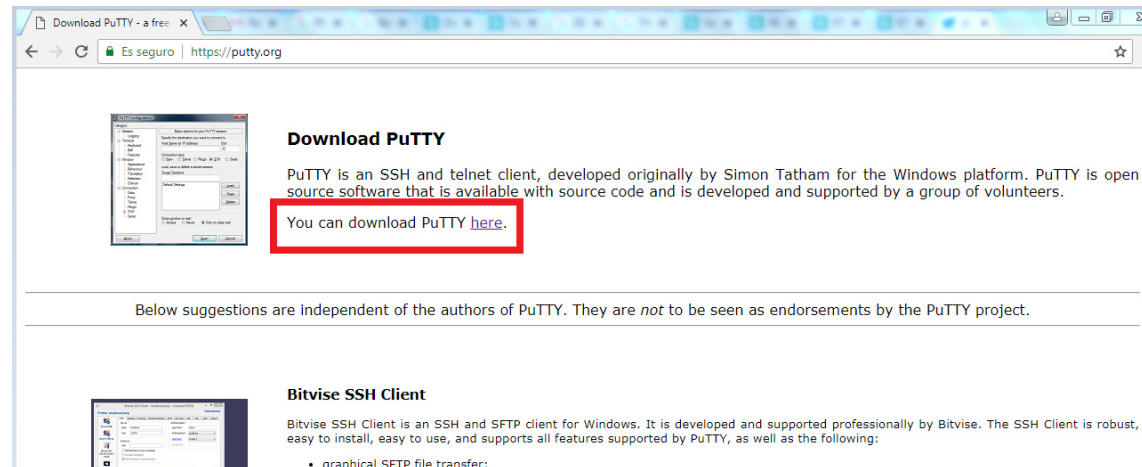


Figura 43: Lloc de descarga del PuTTY.

S'obrirà la següent pestanya en la que ens permetrà descarregar el programa segons el número de bits del nostre sistema operatiu, per comprovar quants bits té el seu ordinador, anar a→ **Panel de control\Sistema y Seguridad\Sistema** i aquí mirar el tipus de sistema.

Download PuTTY: latest release (0.70)

[Home](#) | [FAQ](#) | [Feedback](#) | [Licence](#) | [Updates](#) | [Mirrors](#) | [Keys](#) | [Links](#) | [Team](#)
Download: [Stable](#) | [Snapshot](#) | [Docs](#) | [Changes](#) | [Wishlist](#)

This page contains download links for the latest released version of PuTTY. Currently this is 0.70, released on 2017-07-08.

When new releases come out, this page will update to contain the latest, so this is a good page to bookmark or link to. Alternatively, here is a [permanent link to the 0.70 release](#).

Release versions of PuTTY are versions we think are reasonably likely to work well. However, they are often not the most up-to-date version of the code available. If you have a problem with this release, then it might be worth trying out the [development snapshots](#), to see if the problem has already been fixed in those versions.

Package files

You probably want one of these. They include all the PuTTY utilities.
(Not sure whether you want the 32-bit or the 64-bit version? Read the [FAQ entry](#).)

MSI ("Windows Installer")			
32-bit:	putty-0.70-installer.msi	(or by FTP)	(signature)
64-bit:	putty-64bit-0.70-installer.msi	(or by FTP)	(signature)
Unix source archive			
.tar.gz:	putty-0.70.tar.gz	(or by FTP)	(signature)

Figura 44: Arxiu descarregat del PuTTY

Un cop descarregat, instal·lar seguint les indicacions que proporciona el programa.

2n) Un cop instal·lat, hem de saber en quin port *com* està connectat l'Arduino. Per veure-ho, hem d'anar a *Mi equipo*, fer clic amb el boto dret, i seleccionar l'opció Administrar.

S'obrirà una nova finestra a la qual hem de seleccionar *Administración de dispositivos*, es desplegarà un menú en pantalla en el que hem de seleccionar *puertos COM* i LP, allà ens indicarà el port COM en el que està l'Arduino.

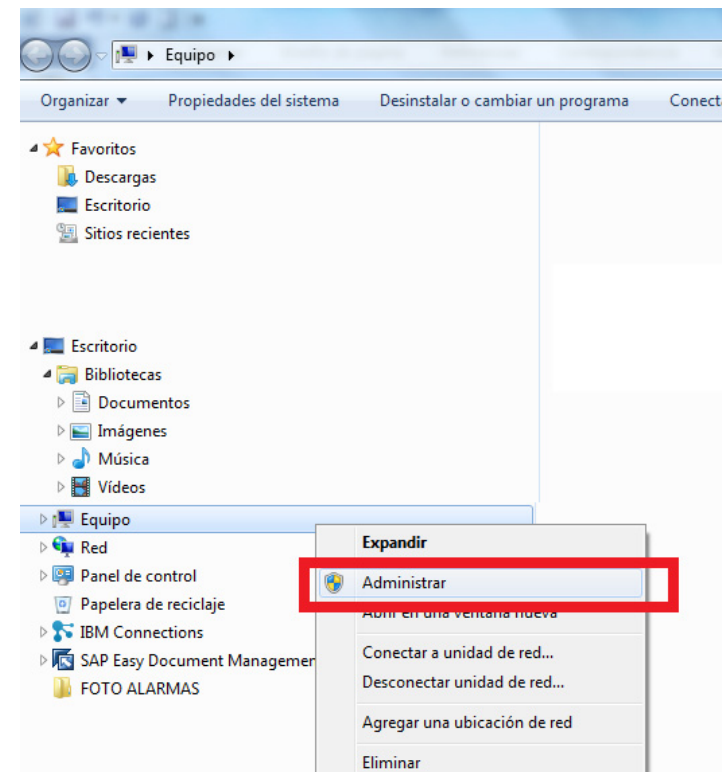


Figura 45: Pas 1 de la configuració del PuTTY

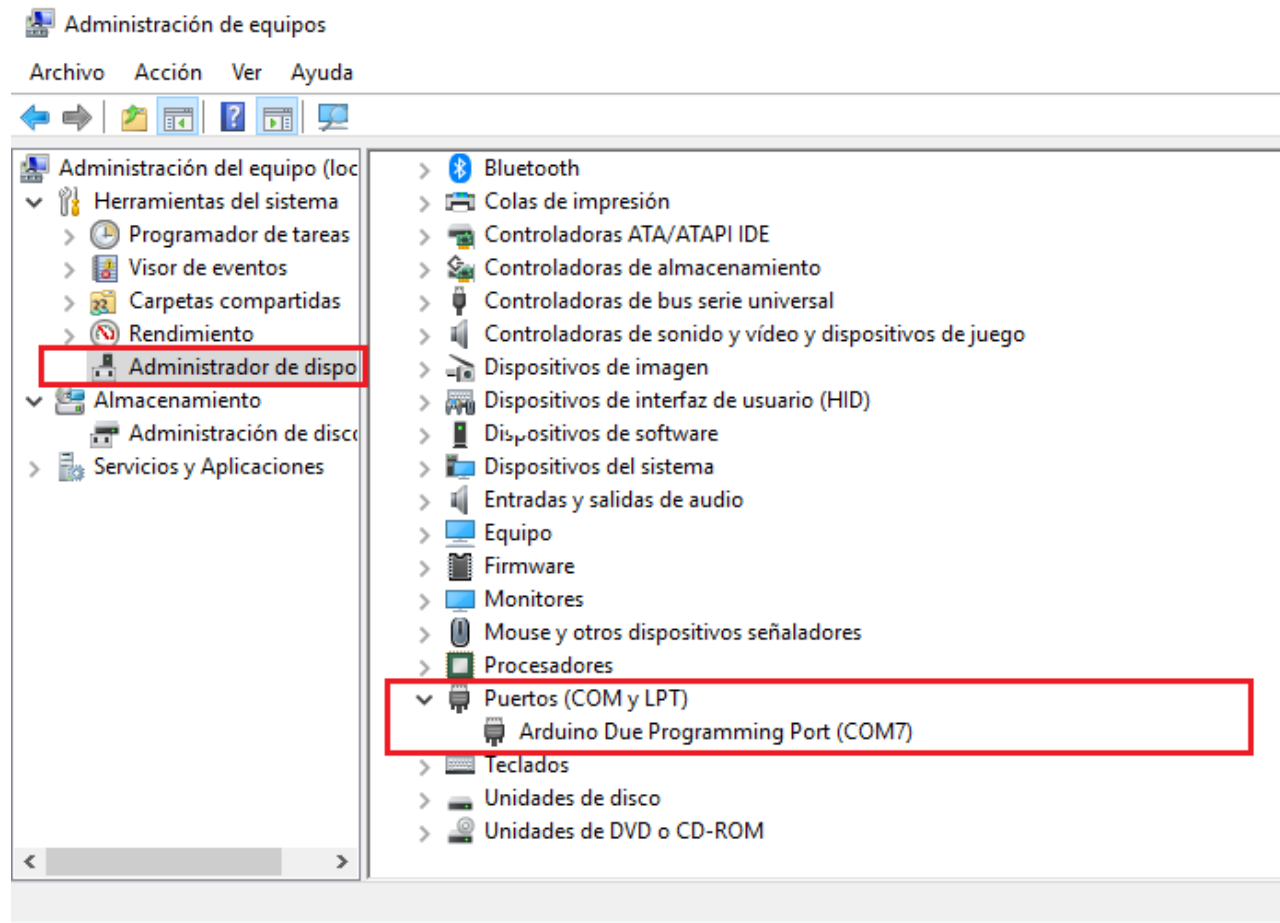


Figura 46: Pas 2 de la configuració del PuTTY

3r) Un cop sabem a quin port està connectat l'Arduino, executem el programa PUTTY.

Un cop obert el PUTTY, seleccionem SERIAL, després posem el COM on estava situat l'Arduino com hem vist al punt anterior.

Assegurem la velocitat igual que en el programa Arduino, en el nostre cas 9600.

Per no haver de seleccionar cada vegada el COM, la velocitat etc. Hi ha la opció de guardar el perfil de l'Arduino, introduint un nom per al perfil que hem de guardar (per exemple ARDUINO DUE), i fent clic a la opció "SAVE".

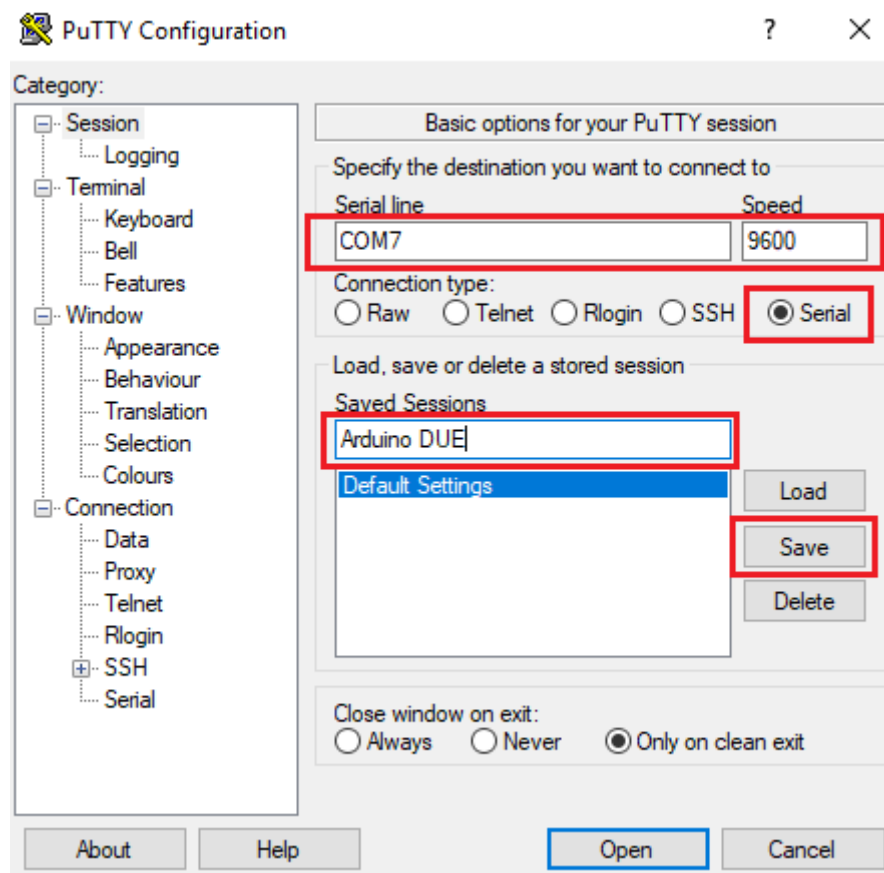
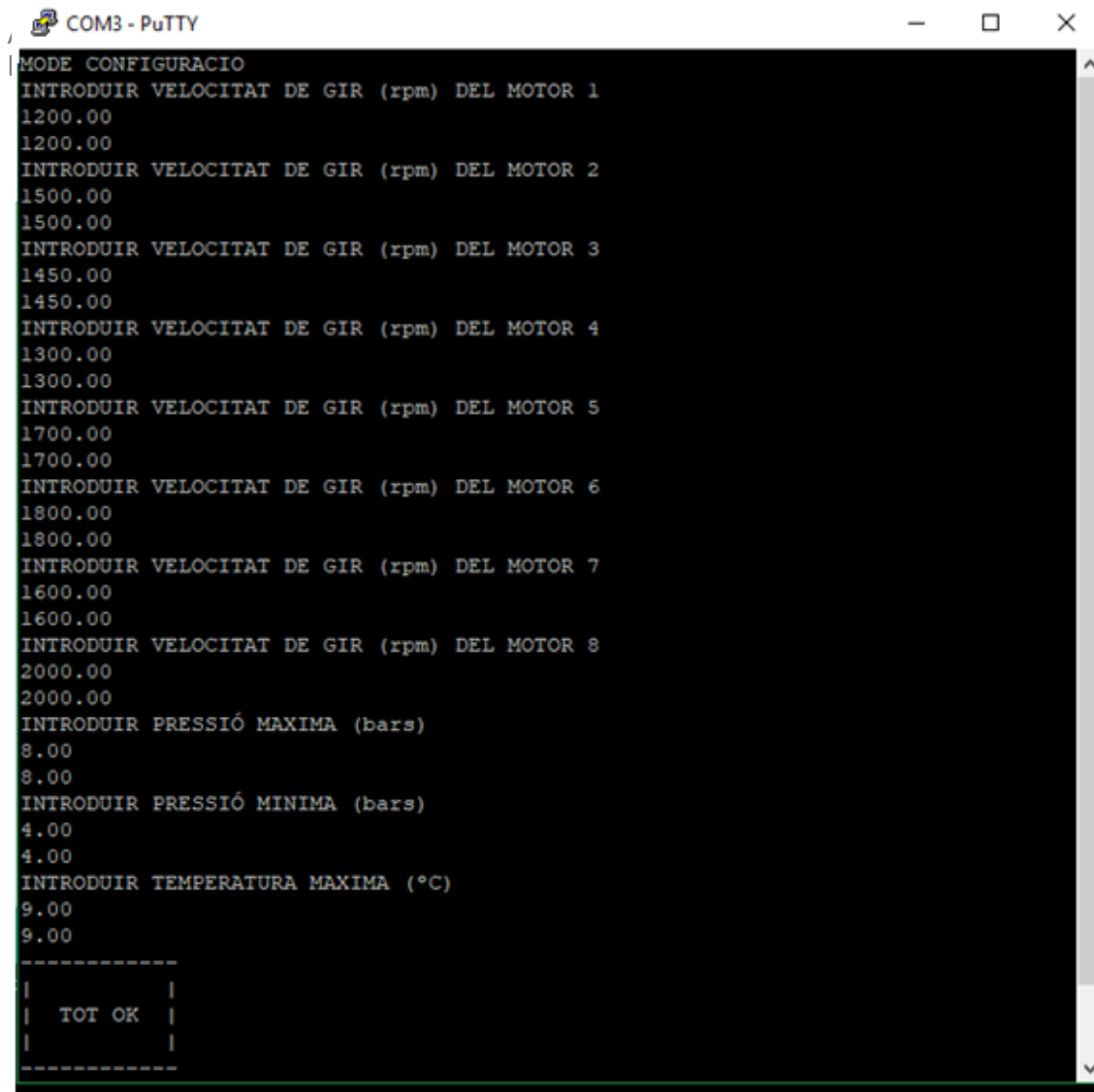


Figura 47: Guardat del perfil de l'Arduino



```
COM3 - PuTTY
MODE CONFIGURACIO
INTRODUIR VELOCITAT DE GIR (rpm) DEL MOTOR 1
1200.00
1200.00
INTRODUIR VELOCITAT DE GIR (rpm) DEL MOTOR 2
1500.00
1500.00
INTRODUIR VELOCITAT DE GIR (rpm) DEL MOTOR 3
1450.00
1450.00
INTRODUIR VELOCITAT DE GIR (rpm) DEL MOTOR 4
1300.00
1300.00
INTRODUIR VELOCITAT DE GIR (rpm) DEL MOTOR 5
1700.00
1700.00
INTRODUIR VELOCITAT DE GIR (rpm) DEL MOTOR 6
1800.00
1800.00
INTRODUIR VELOCITAT DE GIR (rpm) DEL MOTOR 7
1600.00
1600.00
INTRODUIR VELOCITAT DE GIR (rpm) DEL MOTOR 8
2000.00
2000.00
INTRODUIR PRESSIO MAXIMA (bars)
8.00
8.00
INTRODUIR PRESSIO MINIMA (bars)
4.00
4.00
INTRODUIR TEMPERATURA MAXIMA (°C)
9.00
9.00
-----
| TOT OK |
|-----|
```

Figura 48: Mode de configuració

Per connectar l'Arduino, tan sols em de fer clic al botó de OPEN, i se'ns obrirà una pantalla en negre en la qual apareixerà el missatge de introduir els valors màxims de cada element mòbil, i em d'anar introduint els valors desitjats. Un cop teclejat el nombre que desitgem polsar ENTER, un cop haguem introduït tots els valors, hi haurà un missatge de TOT OK.

A mesura que anem introduint els valors aquests apareixeran dues vegades per pantalla per confirmar que sigui correcte.

Si es desitgés modificar un dels valors, el mode de configuració torna al inici un cop han transcorregut 10 segons, i s'han de introduir de nou totes les variables.

3.Funcionament de les Alarmes.

El sistema d'alarma consta de una alarma sonora generada per un Buzzer, i una visual que apareixerà al monitor.

Les alarmes funcionen de la següent manera:

Velocitat:

Si la velocitat proporcionada pels sensors capacitius és inferior a la velocitat introduïda per l'usuari en el mode de configuració, no hi ha cap alarma, ni sonora ni visual.

Per altre banda, si la velocitat mesurada és superior al límit establert per l'usuari en el mode de configuració apareixerà un missatge a la pantalla, indicant quin motor supera el límit establert. Sonarà una alarma durant 10 segons per cridar la atenció de l'usuari.

Temperatura:

L'alarma de temperatura funciona de la següent manera: si el valor llegit pel sensor és superior durant 30 segons al introduït per l'usuari en el mode de configuració, apareixerà un missatge a la pantalla, indicant que s'ha superat el límit de temperatura, indicarà la temperatura actual d'aquest valor i sonarà una alarma durant 10 segons, per cridar la atenció de l'usuari.

Pressió:

L'alarma de la pressió funciona seguint les comparacions anteriors, però en aquest cas ha de funcionar en un rang de valors, dels quals l'usuari ha d'introduir el valor mínim i el valor màxim de pressió en bars. Per pantalla es visualitzarà el valor actual de pressió, apart dels límits superior i inferior. Si es supera un dels límits introduïts per l'usuari, apareixerà un missatge per pantalla indicant. Si hi ha baixa pressió o, pel contrari, hi ha una pressió molt alta, també sonarà una alarma durant 5-10 segons, per cridar la atenció de l'usuari.

Les alarmes es mostraran per pantalla com a l'exemple següent.



Figura 49: Imatge del monitor, amb el missatge que visualitzarà l'usuari

4.Programa Arduino

Com configurar el programa Arduino per modificar el programa.

1r) Per descarregar el programa ARDUINO IDE (En cas de ja tenir instal·lat el PUTTY, procedeixi al pas 2).

Anar a la web : <https://www.arduino.cc/en/Main/Software> i descarregar el programa, fent clic on s'indica a la fotografia següent:

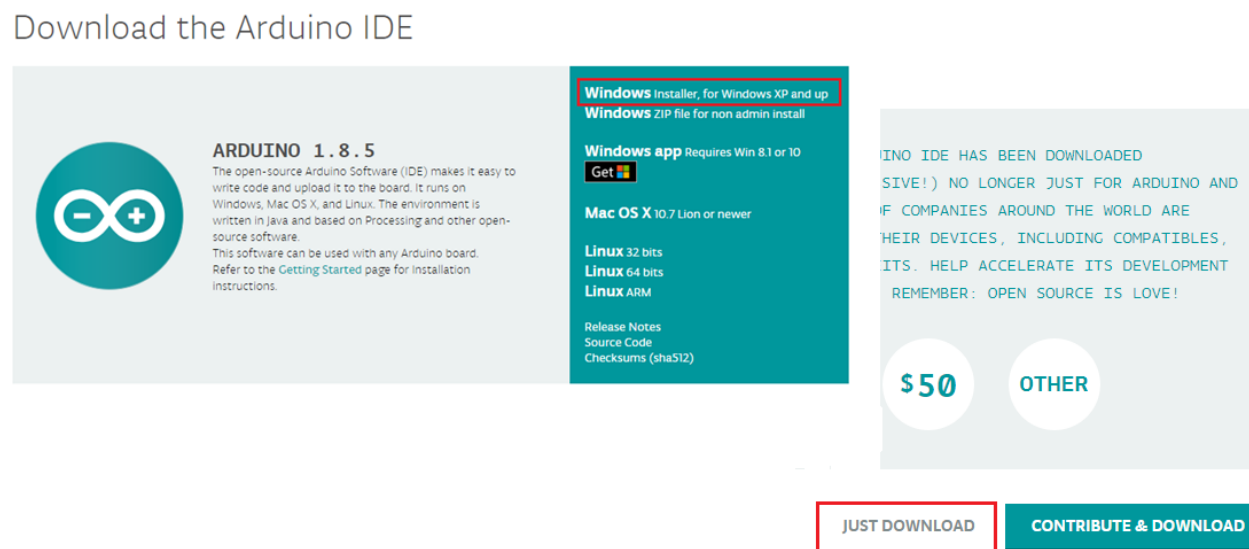


Figura 50: Pàgina web per descarregar l'Arduino

2n) Un cop instal·lat, el programa Arduino IDE, s'ha de configurar la placa al menú *Herramientas* tal com es mostra a la fotografia següent. Per saber el port COM on està connectat l'Arduino s'ha de revisar el punt anterior (configuració del PuTTY).

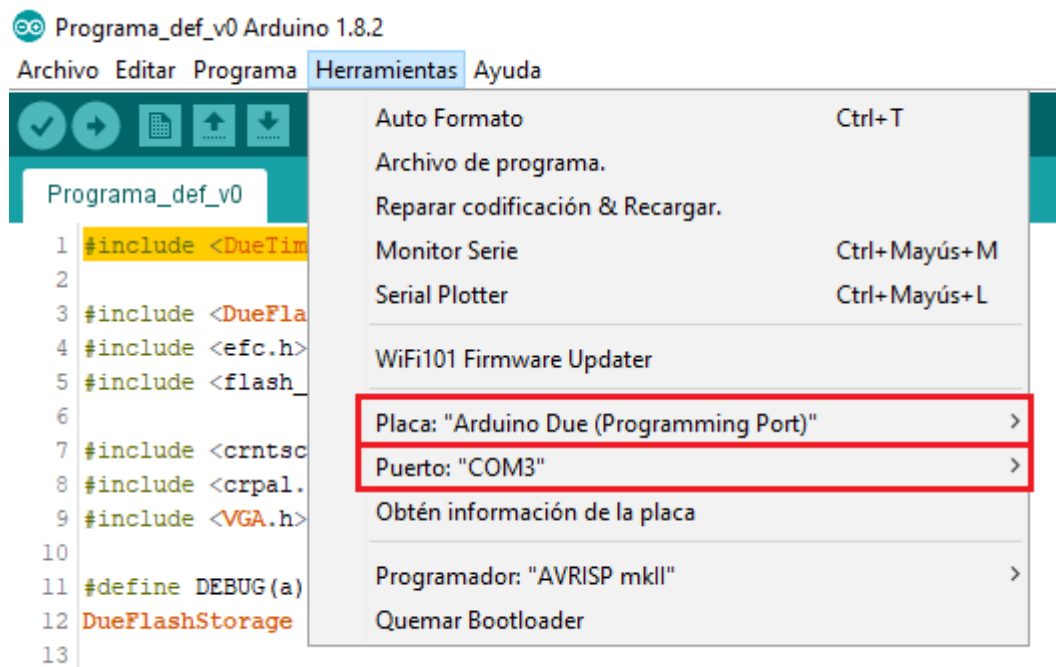


Figura 51: Configuració de la placa al programa

3r) Per comprovar el si el programa és correcte s'ha de compilar mitjançant el botó numero 1. I per carregar el programa a la placa s'ha de polsar el icona numero 2.

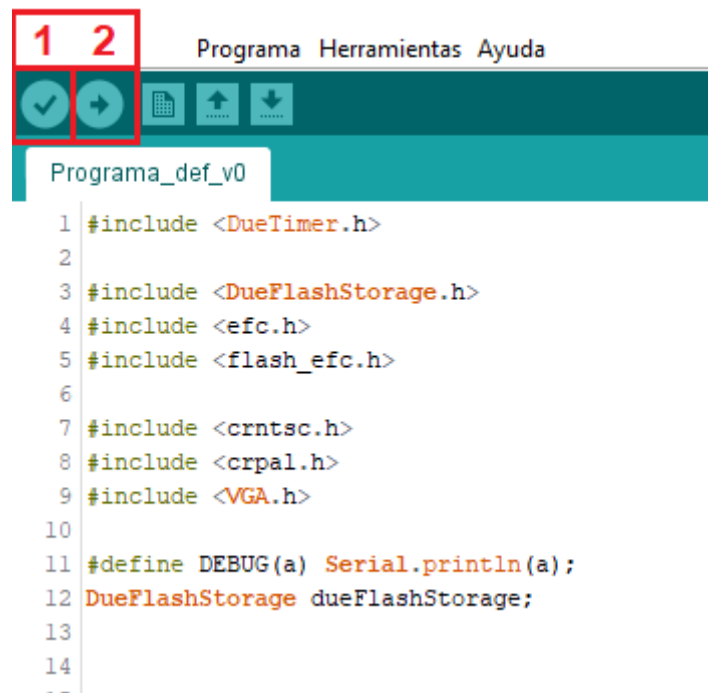


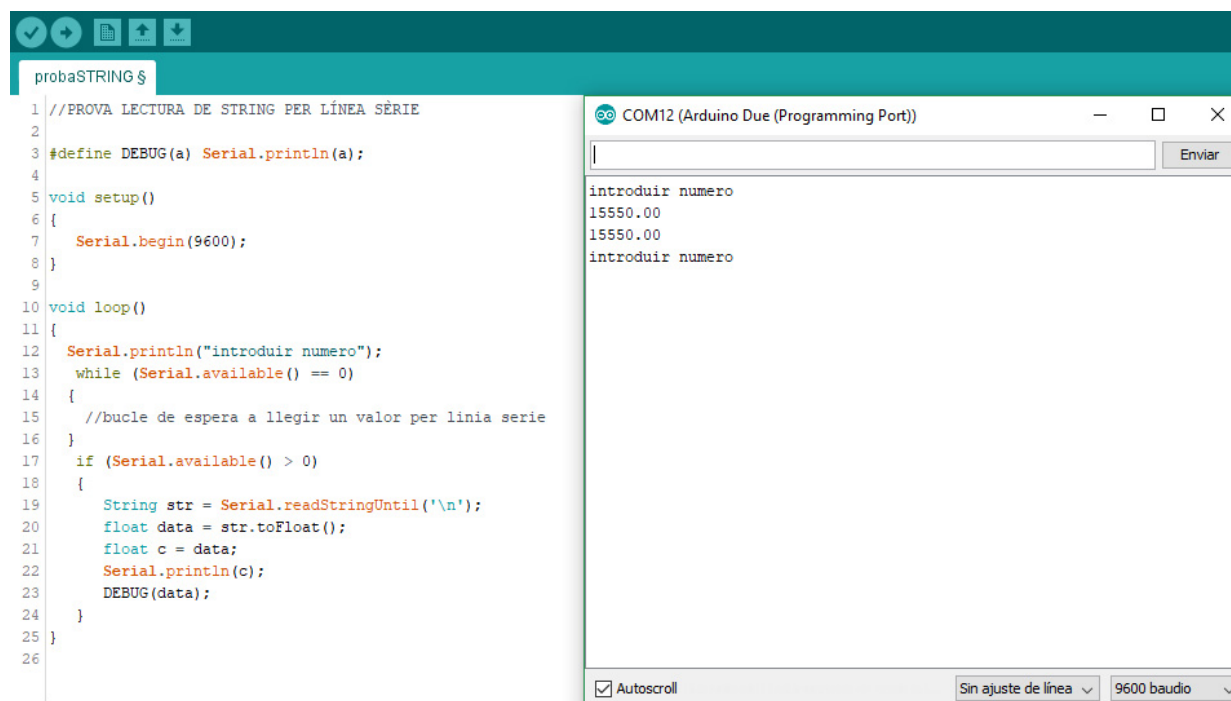
Figura 52: Iconos de compilar i cargar el programa

7. TEST DEL SISTEMA

7.1. PROGRAMA

7.1.1. MODE CONFIGURACIÓ

Previ a la connexió entre l'Arduino i el client PuTTY, hem realitzat diverses proves per a la lectura de les dades posades en la línia sèrie. Degut a que la línia sèrie, llegeix caràcter a caràcter, vam realitzar aquesta funció per llegir una cadena de dades.



```
1 //PROVA LECTURA DE STRING PER LÍNEA SÈRIE
2
3 #define DEBUG(a) Serial.println(a);
4
5 void setup()
6 {
7   Serial.begin(9600);
8 }
9
10 void loop()
11 {
12   Serial.println("introduir numero");
13   while (Serial.available() == 0)
14   {
15     //bucle de espera a llegir un valor per linia serie
16   }
17   if (Serial.available() > 0)
18   {
19     String str = Serial.readStringUntil('\n');
20     float data = str.toFloat();
21     float c = data;
22     Serial.println(c);
23     DEBUG(data);
24   }
25 }
26
```

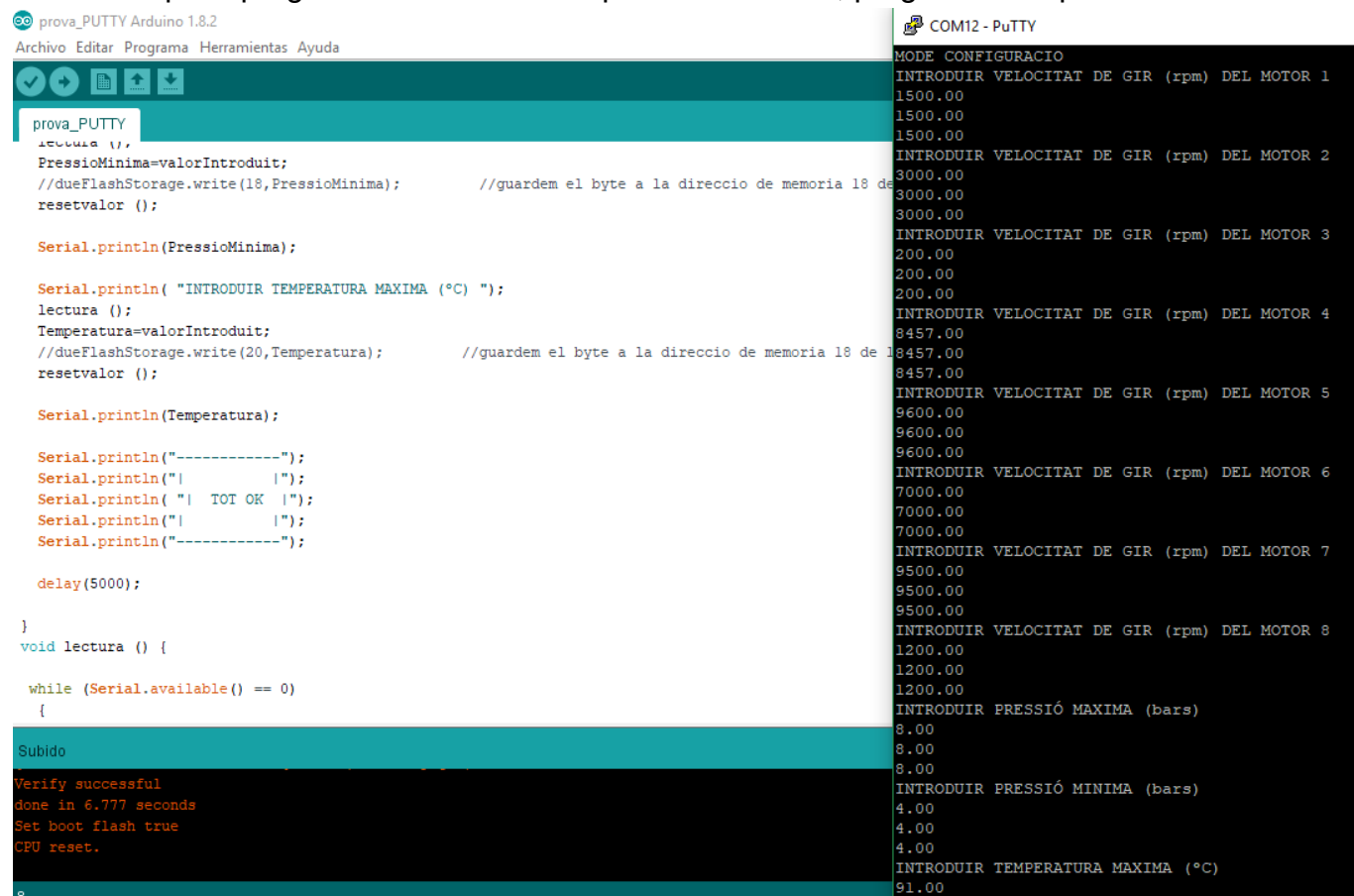
COM12 (Arduino Due (Programming Port))

introduir numero
15550.00
15550.00
introduir numero

☒ Autoscroll Sin ajuste de línea 9600 baudio

Figura 53: Programa de prova de lectura.

Per fer les proves de connexió mitjançant l'Arduino amb el client PuTTY, es va realitzar un programa per enviar les dades per línia sèrie i llegir-les al PuTTY. Aquest programa està inclòs a l'apartat d'annexes, programes de prova.



The image shows two windows side-by-side. The left window is the Arduino IDE, titled 'prova_PUTTY Arduino 1.8.2'. It contains a C++ program for testing serial communication. The program defines a 'lectura()' function that reads data from the serial port and prints it to the serial monitor. It also includes a 'void setup()' function that initializes the serial port and a 'void loop()' function that calls 'lectura()' and prints the results. The right window is a PuTTY terminal window, titled 'COM12 - PuTTY'. It displays the output of the Arduino program, showing the results of the 'lectura()' function calls. The output includes the text 'INTRODUIR VELOCITAT DE GIR (rpm) DEL MOTOR 1' followed by the value '1500.00', and so on for motors 2 through 8. It also shows the text 'INTRODUIR PRESSIÓ MAXIMA (bars)' followed by the value '8.00', and 'INTRODUIR PRESSIÓ MINIMA (bars)' followed by the value '4.00'. Finally, it shows 'INTRODUIR TEMPERATURA MAXIMA (°C)' followed by the value '91.00'.

```
prova_PUTTY
lectura ();
PressioMinima=valorIntroduit;
//dueFlashStorage.write(18,PressioMinima);      //guardem el byte a la direccio de memoria 18 de
resetvalor ();

Serial.println(PressioMinima);

Serial.println( "INTRODUIR TEMPERATURA MAXIMA (°C) ");
lectura ();
Temperatura=valorIntroduit;
//dueFlashStorage.write(20,Temperatura);      //guardem el byte a la direccio de memoria 18 de 1
resetvalor ();

Serial.println(Temperatura);

Serial.println("-----");
Serial.println("|          |");
Serial.println( " | TOT OK | ");
Serial.println("|          |");
Serial.println("-----");

delay(5000);

}

void lectura () {

while (Serial.available() == 0)
{

Subido
Verify successful
done in 6.777 seconds
Set boot flash true
CPU reset.
8
```

```
COM12 - PuTTY
MODE CONFIGURACIO
INTRODUIR VELOCITAT DE GIR (rpm) DEL MOTOR 1
1500.00
1500.00
1500.00
INTRODUIR VELOCITAT DE GIR (rpm) DEL MOTOR 2
3000.00
3000.00
3000.00
INTRODUIR VELOCITAT DE GIR (rpm) DEL MOTOR 3
200.00
200.00
200.00
INTRODUIR VELOCITAT DE GIR (rpm) DEL MOTOR 4
8457.00
8457.00
8457.00
INTRODUIR VELOCITAT DE GIR (rpm) DEL MOTOR 5
9600.00
9600.00
9600.00
INTRODUIR VELOCITAT DE GIR (rpm) DEL MOTOR 6
7000.00
7000.00
7000.00
INTRODUIR VELOCITAT DE GIR (rpm) DEL MOTOR 7
9500.00
9500.00
9500.00
INTRODUIR VELOCITAT DE GIR (rpm) DEL MOTOR 8
1200.00
1200.00
1200.00
INTRODUIR PRESSIÓ MAXIMA (bars)
8.00
8.00
8.00
INTRODUIR PRESSIÓ MINIMA (bars)
4.00
4.00
4.00
INTRODUIR TEMPERATURA MAXIMA (°C)
91.00
```

Figura 54: Programa prova de connexió amb el client PuTTY.

7.1.2. MODE RUN

Per a la prova del sensor de Temperatura i dels sensor de pressió, es va utilitzar el mateix codi, amb petites variacions. Tan sols canvien els paràmetres de les rectes de voltatge-temperatura i voltatge-pressió. Amb aquest codi vam comprovar que l'Arduino llegia correctament la temperatura ambiental. A més, també varem comprovar que al escalfar aquest, el valor també variava correctament.

```
1 int sensor_temperatura = A2;
2 int sensor_pressio = A3;
3
4 int Vt, Temperatura, pendent, Ct;
5 int Vp, Pressio,pendent2, Cp;
6
7 void setup() {
8   Serial.begin(9600);
9   pinMode(sensor_temperatura,INPUT);
10  pinMode(sensor_pressio, INPUT);
11 }
12
13 void loop () {
14   //CAS SENSOR TEMPERATURA
15
16   //pasem el voltatge rebut de la Ni200 a bits
17   Vt = (analogRead(A2)/1023.0)*3.3; // (bits/2^n-1)*Vmax
18   // voltatge a temperatura
19   Temperatura = Vt*pendent+Ct; //y=mx+c
20
21   //CAS SENSOR PRESSIÓ
22
23   //pasem el voltatge rebut del sensor de pressio a bits
24   Vt = (analogRead(A3)/1023.0)*3.3; // (bits/2^n-1)*Vmax
25   // voltatge a pressio
26   Pressio = Vp*pendent2+Cp; //y=mx+c
27 }
```

Figura 55: Programa prova dels sensors de Temperatura i Pressió.

7.2. CONEXIONS I SENSORS

Per el que fa als comptadors de revolucions es va fer la prova mitjançant dues plaques Arduino. Una emetia una senyal de parpelleig d'un LED i l'altre Arduino, mitjançant un sistema de interrupcions i de temporitzadors, contarà la freqüència del parpelleig i la convertirà a revolucions per minut. El codi de tots dos Arduinos, està adjunt al apartat d'annexes.

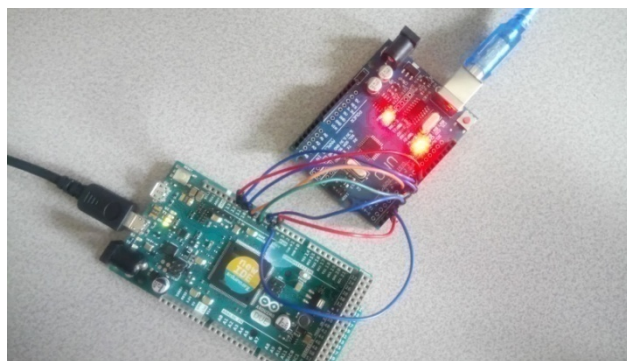


Figura 56: Connexions del programa prova del compta revolucions.

Per la prova d'enviar dades a la pantalla mitjançant un cable VGA, varem connectar els pins necessaris (MOSI, D43, D42 i GND) a una *protoboard* per poder enllaçar-los amb el cable VGA mitjançant les resistències esmentades anteriorment. Pel que fa al programa, vam escriure un codi per tal que es mostressin sempre per pantalla totes les alarmes. A part es mostren les revolucions actuals i el límit establert al mode de configuració. Aquestes dades són experimentals ja que estàvem fent proves per la correcta visualització.



Figura 57: Montatge del monitor



Figura 58: Imatge dels missatges que sortiran pel monitor

8. PRESSUPOST ECÒNOMIC

Un cop sabem quins són els sensors que necessitem, els circuits que els complementen i faran possible l'adquisició de dades i l'Arduino que necessitem, podem fer un pressupost econòmic. Per fer-ho s'han tingut en compte tan el material com la ma d'obra.

Material	Preu/u (€)	Quantitat	Preu (€)
Resistència 150K	0,18	2	0,36
Resistència 100K	0,91	1	0,91
Resistència 15K	0,06	2	0,12
Resistència 10K	0,13	24	3,12
Resistència 3K	0,07	1	0,07
Resistència 82 Ohms	0,07	2	0,14
Resistència 100 Ohms	0,12	1	0,12
Resistència 330 Ohms	0,12	4	0,48
Led	0,42	3	1,26
Interruptor	0,94	2	1,88
Condensadors (10uF)	0,18	2	0,36
Condensador (1uF)	0,16	1	0,16
Amplificador UA741	0,424	2	0,848
MAX 233	5,43	1	5,43
Potenciòmetre multivolta (50K)	0,63	1	0,63
Sòcol20p.	0,42	1	0,42
Sòcol 8p	0,17	2	0,34

Material	Preu/u (€)	Quantitat	Preu (€)
ConnectorfemellaDB9	0,76	1	0,76
Entrada DB9 (Connector)	1,49	1	1,49
ConnectorDB15 (VGA)	2,64	1	2,64
Buzzer	1,02	1	1,02
Transistor M140	0,36	1	0,36
Estabilització de tensió7805	0,22	1	0,22
Transistor BC108	0,33	8	2,64
Pins de Soldadura (40p)	1,33	2	2,66
Header femella (40p)	1,86	2	3,72
Regleta per PCB (3p)	1,67	1	1,67
Regleta per PCB (2p)	1,2	10	12
ArduinoDue	34	1	34
Sensor Inductiu	39,47	8	315,76
Sensor Temperatura (Ni200)	3,67	1	3,67
Sensor pressió (WIKA-A10)	135	1	135
Placa impresa	50	1	50
Monitor (TV)	78,98	1	58,98
TOTAL			643,24

Per altra banda, per poder dur a terme aquest projecte s'estima que son necessàries unes 80 hores de feina, les quals tindrien un preu de 25€/h. Per tan, la ma d'obra tindria un preu de 2.000 €.

PRESSUPOST TOTAL: 2643,24 €

9. CONCLUSIONS

Recordem que el propòsit d'aquest projecte era la implementació d'un sistema de sensors a una collitadora del model Laverda M84, mitjançant un sistema Arduino que ens permeti detectar quan la màquina falla.

Una part important del projecte ha sigut el disseny dels circuits i la placa PCB. Per la creació de la PCB, la complexitat residia en què no s'havia dissenyat cap placa PCB, ni s'havia utilitzat el programa Altium, i que per tant es va haver d'aprendre a fer PCB de manera autodidàctica.

Per altra banda, el programa Arduino ha tingut diverses complicacions a causa del model d'Arduino que s'havia elegit, Arduino Due. Aquest model no disposa de llibreries per modificar ni guardar a la memòria EPROM. Tampoc porta llibreries de sèrie per controlar els diferents timers. Finalment s'han trobat les alternatives adients controlar els timers i per guardar variables no volàtils, com ha sigut l'ús de la memòria Flash.

A dia d'avui, el sistema de sensorat que hem realitzat funciona correctament en el laboratori, però aquest no s'ha pogut provar en l'entorn de treball en què estarà, encara que, pel seu correcte funcionament en les proves de laboratori i simulació, podríem assegurar que aquest funcionarà quan estigui implementat a la collitadora.

De cara a millores del sistema de sensorat creiem que seria més adequat utilitzar components electrònics diferents per intentar obtenir un senyal millor a l'arduino procedent dels sensors. Un exemple seria utilitzar altres amplificadors, com el INA128, en comptes del Ua741.

10. AGRAÏMENTS

Volem agrair al nostre Tutor de TFG, Jaume Miret, per haver-nos donat l'oportunitat de dur a terme aquest projecte, ajudar-nos en tots els dubtes que ens anaven sorgint i, sobretot, per confiar amb nosaltres. També volem donar les gràcies al Xavi Rosselló, propietari de la collitadora.

Per altra banda, volem agrair a les nostres famílies, per ser-hi sempre, aguantar els mals dies i ,sobretot, per ser el principal pilar duran aquests anys de grau.

Finalment, volem agrair a totes aquelles persones que d'una manera o altre, ens han ajudat a poder dur a terme el projecte i, en especial, al Francisco Fernandez, cap de torn de manteniment a Mahle S.A.

11. BIBLIOGRAFIA

11.1 ENTRADES DE FÒRUM:

[1] Stimmer. [Entrada de fòrum]. A: VGA library – nowwith TV output. [En línia]. 13 maig 2013. Disponible a: <<https://forum.arduino.cc/index.php?topic=150517.msg1129884#msg1129884>>.

[2] Mlikesit. [Entrada de fòrum]. A: 3 WireThermistorWiring. [En línia]. 06 febrer 2014. Disponible a: <<http://forum.arduino.cc/index.php?topic=215788.0>>.

[3] FernandoGarcia. [Entrada de fòrum]. A: EEPROM. [En línia]. 22 octubre 2012. Disponible a: <<http://forum.arduino.cc/index.php?topic=128498.0>>.

[4] R.Pieter. [Entrada de fòrum]. A: Store andchange data on Due wicharenotResetafterpowercycle. [En línia]. 31 març 2014. Disponible a: <<http://forum.arduino.cc/index.php?topic=229908.0>>.

[5] M3741. [Entrada de fòrum]. A: New ArduinoDue Library: DueEEPROM. [En línia]. 02 octubre 2013. Disponible a: <<http://forum.arduino.cc/index.php?topic=191298.0>>.

[6] User1529034. [Entrada de fòrum]. A: New Nickel 500 and 200 ohms at 0 Celsius. [En línia]. 17 març 14 Disponible a: <<https://electronics.stackexchange.com/questions/103183/nickel-500-and-200-ohms-at-0-celsius>>.

11.2 ENTRADES DE BLOG:

[7] Víctor Ventura. Control de dispositivossérie UART desde el ordenador con PuTTY. [Entrada de blog]. [En línia]. 14 desembre

2014. Disponible a: <<https://polaridad.es/usb-serie-terminal-putty/>>.

[8] Scott, Paul. Arduino TV out cable. A: Project HUB.[Entrada de blog]. [En línia]. 15 setembre 2015. Disponible a: <<https://create.arduino.cc/projecthub/paulscott56/arduino-tv-out-cable-64de69>>.

[9] LuisLLamas. Guardar Variables en Arduino y la memoria no volátil EEPROM. [Entrada de blog]. [En línia]. 09 octubre 2016. Disponible a: <<https://www.luisllamas.es/guardar-variables-entre-reinicios-con-arduino-y-la-memoria-no-volatil-eeeprom/>>.

11.3 E-BOOK:

[10] SchmidtMaik .Ed. Arduino a QuickStartGuide. [En línia]. Disponible a: <http://www.multimedialab.be/doc/erg/2017-2018/Arduino_A_Quick-Start_Guide_2nd_Edition_The_Pragmatic_Programmers_2015.pdf>.

11.4 VÍDEOS:

[11] AvE. 4-20mA Industrial Sensor + Arduino. [Vídeo en línia]. Categoria: ciència i tecnologia. Llicència: estàndard de Youtube. 21 juny 2017. Disponible a: <<https://www.youtube.com/watch?v=6di24oldlSs>>.

11.5 ARXIUS DESCARREGABLES:

[12] Due VGA library. [Arxiu descarregable]. Versió: V0.987654321. Disponible a: <<https://github.com/stimmer/DueVGA/tree/master/VGA>>, 23 novembre 2013.

[13] Timer Library fully implemented for Arduino DUE. [arxiu descarregable]. Versió: V2.0. Disponible a: <<https://github.com/ivanseidel/DueTimer>>, 09 juny 2013.

[14] Due Flash Storage library. [Arxiu descarregable]. Versió: V2.3. Disponible a: <<https://github.com/sebnil/DueFlashStorage>>13 novembre 2013.

[15] Due_eepromlibrary. [Arxiu descarregable]. Versió: V1.5.4. Disponible a: https://github.com/m3741/due_eeprom.02 octubre 2013.

[16] WIKA a-10 datasheet. [Arxiu descarregable]. Versió: V1. Disponible a: <https://docs-emea.rs-online.com/webdocs/131f/0900766b8131f667.pdf>

[17] Inductive proximity sensor IME12-04bpszc0s datasheet.[arxiu descarregable]. Versió: V1. Disponible a: <https://www.sick.com/ag/en/proximity-sensors/inductive-proximity-sensors/ime/ime12-04bpszc0s/p/p228481>.

[18] 200° C series nickel sensor with wires.[arxiu descarregable]. Versió: V1. Disponible a: https://www.ist-ag.com/sites/default/files/DTN200_E.pdf.

[19] Software PuTTY. [arxiu descarregable]. Versió: 0.70. Disponible a: <<https://putty.org/>>

[20] Ministeri d'Espanya. Coseñadoras de granos i semillas. [Arxiu descargable] . Disponible a: <http://www.mapama.gob.es/es/ministerio/servicios/informacion/cosech-granos-semillas_tcm30-58361.pdf>

[21] RS. *Sensor de temperatura, PT100*. [En línea] <<https://es.rs-online.com/web/p/sensores-de-temperatura-con-resistencia-de-platino/3629799/>>

- [22] RS. *Sensor de temperatura termopar tipus K*. [En línia] <<https://uk.rs-online.com/web/p/thermocouples/3630294/>>
- [23] RS. *Sensor de pressió, 3100B0010G01B000*. [En línia] <<https://es.rs-online.com/web/p/sensores-de-presion/4554624/>>
- [24] RS. *Sensor de pressió, WIKA-A10*. [En línia] <<https://uk.rs-online.com/web/p/hydraulic-pressure-sensors/7255948/>>
- [25] RS. *Sensor de pressió, PT5415*. [En línia]. <<https://es.rs-online.com/web/p/sensores-de-presion/8353901/>>
- [26] RS. *Sensor inductiu, NBB5-18GM50-E2-V1*. [En línia] <<https://es.rs-online.com/web/p/sensores-de-proximidad-inductivos/2492366/>>
- [27] Sick, sensor intel·ligent. *Sensores de proximidad inductivos IME*. [En línia]. <<https://www.sick.com/aq/en/proximity-sensors/inductive-proximity-sensors/ime/ime12-04bpszc0s/p/p228481>>
- [28] RS. *Sensor inductiu BES M08EC-PSC20B-S49G*. [En línia]. <<https://es.rs-online.com/web/p/sensores-de-proximidad-inductivos/1243765/>>
- [29] Texas instruments. *Datasheet Ua741*. [Arxiu descarregable]. Disponible en: <<http://www.ti.com/lit/ds/symlink/ua741.pdf>>
- [30] SGS-THOMSON Microelectronics. *Datasheet BC108*. [Arxiu descarregable]. Disponible en: <<https://www.silicon-ark.co.uk/datasheets/bc107-108-datasheet-%20stm.pdf>>
- [31] Radio museum. *Datasheet MC140*. [Arxiu descarregable]. Disponible en: <https://www.radiomuseum.org/tubes/tube_mc140.html>
- [32] Màxim integrated. *MAX233: + Controladores / receptores multicanal RS-232 con alimentación de 5 V*. [Arxiu descarregable]. Disponible en: <<https://www.maximintegrated.com/en/products/interface/transceivers/MAX233.html>>

[33] Products limited. *Datsheet Buzzer 32S4120LF*. [Arxiu descarregable]. Disponible en:
<<https://www.jameco.com/Jameco/Products/ProdDS/138713.pdf>>

[34] On Semiconductor. *Datasheet MC7805* [Arxiu descarregable]. Disponible en:
<<https://www.onsemi.com/pub/Collateral/MC7800-D.PDF>>

11.6 WEBS

[35] Matillas.otg. Así funciona una cosechadora. [En línia]. Disponible en: <<http://www.matillas.org/2014/07/30/asi-funciona-una-cosechadora/>>

[36] Rs-components. Sensors de temperatura: pt100 [En línia] <<https://uk.rs-online.com/web/p/thermocouples/3630294/>>

[37] Rs-components. Sensors de temperatura. [En línia]<<https://es.rs-online.com/web/p/sensores-de-temperatura-con-resistencia-de-platino/3629799/>>

[38] Arduino. Arduino DUE. [En línia] <<https://store.arduino.cc/arduino-due>>

12. FOTOTECA

12.1 FIGURES

FIGURA 1: <<https://www.youtube.com/watch?v=Dqm3le15ewQ>>

FIGURA 2: <<http://www.matillas.org/2014/07/30/asi-funciona-una-cosechadora/>>

FIGURA 3: <<http://www.matillas.org/2014/07/30/asi-funciona-una-cosechadora/>> modificada.

FIGURA 4: Font pròpia.

FIGURA 5: <<https://store.arduino.cc/arduino-due>>

FIGURA 6: <<https://es.rs-online.com/web/p/sensores-de-temperatura-con-resistencia-de-platino/3629799/>>

FIGURA 7: Font pròpia

FIGURA 8: <<https://uk.rs-online.com/web/p/thermocouples/3630294/>>

FIGURA 9: Font pròpia

FIGURA 10: <<https://es.rs-online.com/web/p/sensores-de-presion/8353901/>>

FIGURA 11: <<https://uk.rs-online.com/web/p/hydraulic-pressure-sensors/7255948/>>

FIGURA 12: <<https://es.rs-online.com/web/p/sensores-de-presion/4554624/>>

FIGURA 13: Font pròpia

FIGURA 14: <<https://es.rs-online.com/web/p/sensores-de-proximidad-inductivos/1243765/>>

FIGURA 15: <<https://es.rs-online.com/web/p/sensores-de-proximidad-inductivos/2492366/>>

FIGURA 16: <<https://www.sick.com/ag/en/proximity-sensors/inductive-proximity-sensors/ime/ime12-04bpszc0s/p/p228481>>

FIGURA 17: Font pròpia

FIGURA 18: Font pròpia

FIGURA 19: Font pròpia

FIGURA 20: Font pròpia

FIGURA 21: Font pròpia

FIGURA 22: Font pròpia

FIGURA 23: Font pròpia

FIGURA 24: Font pròpia

FIGURA 25: Font pròpia

FIGURA 26: Font pròpia

FIGURA 27: Font pròpia

FIGURA 28: Font pròpia

FIGURA 29: Font pròpia

FIGURA 30: Font pròpia

FIGURA 31: Font pròpia

FIGURA 32:Font pròpia
FIGURA 33:Font pròpia
FIGURA 34:Font pròpia
FIGURA 35:Font pròpia
FIGURA 36:Font pròpia
FIGURA 37:Font pròpia
FIGURA 38:Font pròpia
FIGURA 39:Font pròpia
FIGURA 40:Font pròpia
FIGURA 41:Font pròpia
FIGURA 42:Font pròpia
FIGURA 43:Font pròpia
FIGURA 44:Font pròpia
FIGURA 45:Font pròpia
FIGURA 46:Font pròpia
FIGURA 47:Font pròpia
FIGURA 48:Font pròpia
FIGURA 49:Font pròpia
FIGURA 50:Font pròpia
FIGURA 51:Font pròpia
FIGURA 52:Font pròpia
FIGURA 53:Font pròpia
FIGURA 54:Font pròpia
FIGURA 55:Font pròpia
FIGURA 56:Font pròpia
FIGURA 57:Font pròpia
FIGURA 58:Font pròpia

12.2 GRÀFIQUES

GRÀFIC 1:Font pròpia
GRÀFIC 2:Font pròpia
GRÀFIC 3:Font pròpia
GRÀFIC 4:Font pròpia
GRÀFIC 5:Font pròpia
GRÀFIC 6: Font pròpia

12. ANNEXES